The Mirai Botnet

Ulysses Butler, Thu Vo, and Tung Thai, Torey Clark

Truman State University
Binary Beasts

The Paper

- Title: *Understanding the Mirai Botnet*

The Paper

- Title: *Understanding the Mirai Botnet*
- *The Proceedings of the 26th USENIX Security Symposium*

The Paper

- Title: *Understanding the Mirai Botnet*
- *The Proceedings of the 26th USENIX Security Symposium*
- This paper explores the Mirai botnet

The Paper

- Title: *Understanding the Mirai Botnet*
- *The Proceedings of the 26th USENIX Security Symposium*
- This paper explores the Mirai botnet
- This botnet was responsible for one of the largest DDoS attacks every recorded.

The Paper

- Title: *Understanding the Mirai Botnet*
- *The Proceedings of the 26th USENIX Security Symposium*
- This paper explores the Mirai botnet
- This botnet was responsible for one of the largest DDoS attacks every recorded.
- The purpose of this paper was to learn about how the botnet worked.

The Paper

- Title: *Understanding the Mirai Botnet*
- *The Proceedings of the 26th USENIX Security Symposium*
- This paper explores the Mirai botnet
- This botnet was responsible for one of the largest DDoS attacks every recorded.
- The purpose of this paper was to learn about how the botnet worked.
- Researchers from a number of institutions reversed engineered it to better understand how it spread

The Paper

- Title: *Understanding the Mirai Botnet*
- *The Proceedings of the 26th USENIX Security Symposium*
- This paper explores the Mirai botnet
- This botnet was responsible for one of the largest DDoS attacks every recorded.
- The purpose of this paper was to learn about how the botnet worked.
- Researchers from a number of institutions reversed engineered it to better understand how it spread
- This paper then proposes reforms that can be made to prevent this kind of attack in the future

Contributions

- Lead Author
  - Zane Ma - University of Illinois Urbana-Champaign

Contributions

- Lead Author
  - Zane Ma - University of Illinois Urbana-Champaign
- This paper had help from many different authors

Contributions

- Lead Author
  - Zane Ma - University of Illinois Urbana-Champaign
- This paper had help from many different authors
  - Manos Antonakakis - Georgia Institute of Technology
  - Tim April - Akamai Technologies
  - Michael Bailey - University of Illinois Urbana-Champaign
  - Matthew Bernhard - University of Michigan
  - Elie Bursztein - Google
  - Jaime Cochran - Cloudflare
  - Zakir Durumeric - University of Michigan
  - J. Alex Halderman - University of Michigan

Contributions Cont.

- Continued...
  - Luca Invernizzi - Google
  - Michalis Kallitsis - Merit Network
  - Deepak Kumar - University of Illinois Urbana-Champaign
  - Chaz Lever - Georgia Institute of Technology
  - Joshua Mason - University of Illinois Urbana-Champaign
  - Damian Menscher - Google
  - Chad Seaman - Akamai Technologies
  - Nick Sullivan - Cloudflare
  - Kurt Thomas - Google
  - Yi Zhou - University of Illinois Urbana-Champaign

Bootstraping

- August 1, 2016: Servers owned by DataWagon began a preliminary scan.

Bootstraping

- August 1, 2016: Servers owned by DataWagon began a preliminary scan.
  - DataWagon is a bulletproof web hosting provider.

Bootstraping

- August 1, 2016: Servers owned by DataWagon began a preliminary scan.
  - DataWagon is a bulletproof web hosting provider.
  - Users are allowed to upload and distribute almost anything using their service.

Bootstraping

- August 1, 2016: Servers owned by DataWagon began a preliminary scan.
  - DataWagon is a bulletproof web hosting provider.
  - Users are allowed to upload and distribute almost anything using their service.
- After this scan, the botnet started infecting computers

Bootstraping

- August 1, 2016: Servers owned by DataWagon began a preliminary scan.
  - DataWagon is a bulletproof web hosting provider.
  - Users are allowed to upload and distribute almost anything using their service.
- After this scan, the botnet started infecting computers
  - 1 minute - 800 infected devices

Bootstraping

- August 1, 2016: Servers owned by DataWagon began a preliminary scan.
  - DataWagon is a bulletproof web hosting provider.
  - Users are allowed to upload and distribute almost anything using their service.
- After this scan, the botnet started infecting computers
  - 1 minute - 800 infected devices
  - 10 minutes - 11,000 infected devices

Bootstraping

- August 1, 2016: Servers owned by DataWagon began a preliminary scan.
  - DataWagon is a bulletproof web hosting provider.
  - Users are allowed to upload and distribute almost anything using their service.
- After this scan, the botnet started infecting computers
  - 1 minute - 800 infected devices
  - 10 minutes - 11,000 infected devices
  - 20 hours - 65,000 infected devices

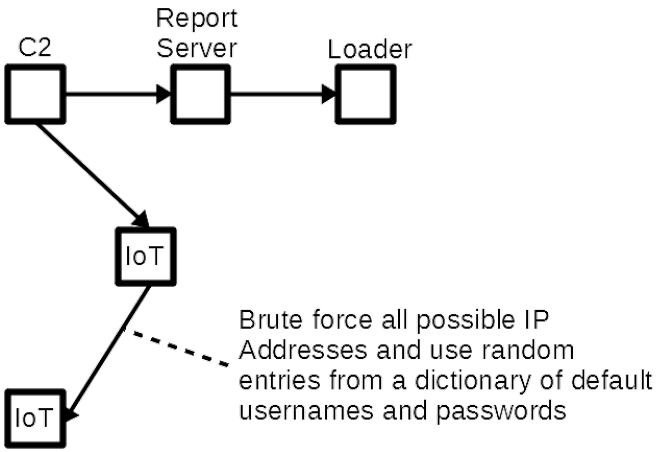
Bootstraping

- August 1, 2016: Servers owned by DataWagon began a preliminary scan.
  - DataWagon is a bulletproof web hosting provider.
  - Users are allowed to upload and distribute almost anything using their service.
- After this scan, the botnet started infecting computers
  - 1 minute - 800 infected devices
  - 10 minutes - 11,000 infected devices
  - 20 hours - 65,000 infected devices
  - Held steady at around 100,000 to 200,000 infections

Bootstraping

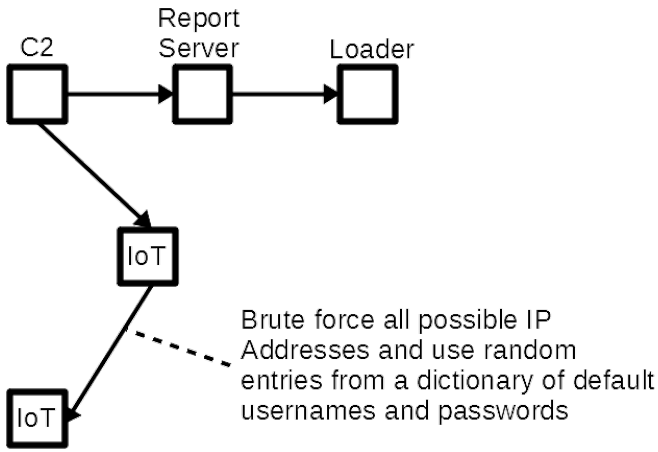
- August 1, 2016: Servers owned by DataWagon began a preliminary scan.
  - DataWagon is a bulletproof web hosting provider.
  - Users are allowed to upload and distribute almost anything using their service.
- After this scan, the botnet started infecting computers
  - 1 minute - 800 infected devices
  - 10 minutes - 11,000 infected devices
  - 20 hours - 65,000 infected devices
  - Held steady at around 100,000 to 200,000 infections
  - In December 2016, it peaked at 600,000 devices before beginning to fade

Spreading



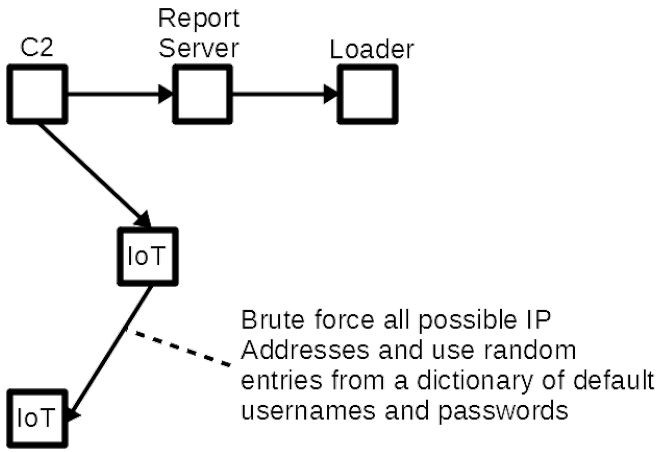
- A member of the botnet begins scanning scanning ports on all IPv4 addresses

Spreading



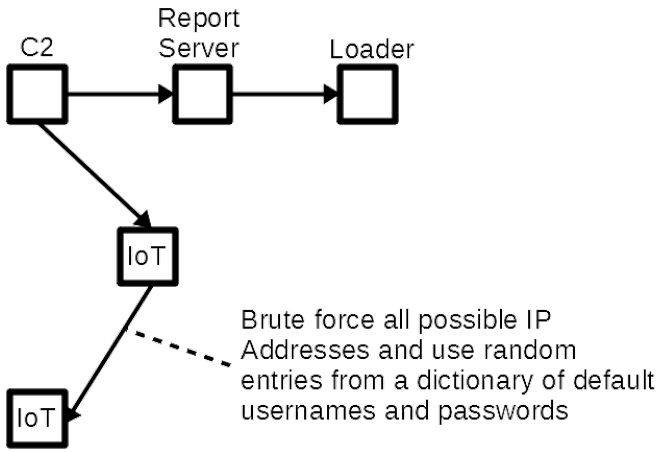
- A member of the botnet begins scanning scanning ports on all IPv4 addresses
- It scans to find open ports for SSH, Telnet, FTP, and other protocols

Spreading



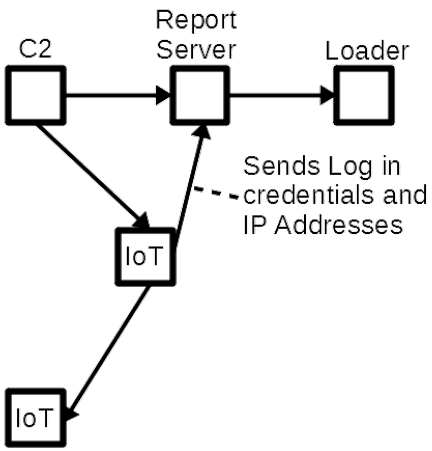
- A member of the botnet begins scanning scanning ports on all IPv4 addresses
- It scans to find open ports for SSH, Telnet, FTP, and other protocols
- It would then use a dictionary attack to brute force into the machine

Spreading



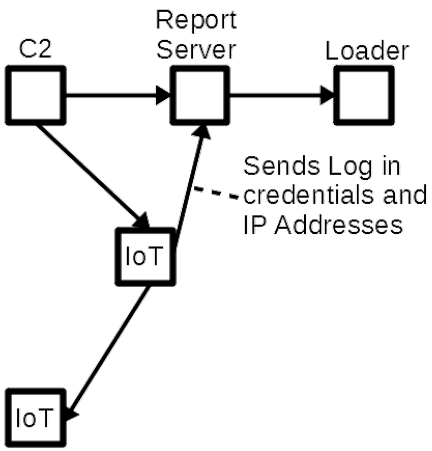
- A member of the botnet begins scanning scanning ports on all IPv4 addresses
- It scans to find open ports for SSH, Telnet, FTP, and other protocols
- It would then use a dictionary attack to brute force into the machine
- These were small dictionaries, containing 60 to about 200 credentials

Spreading



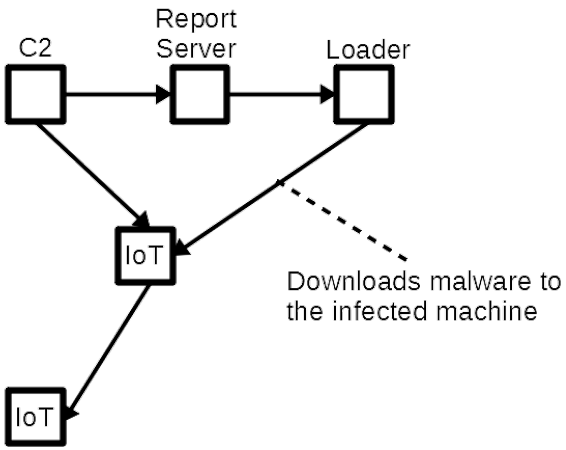
- The address and credentials of the victim machines where then sent to a report server

Spreading



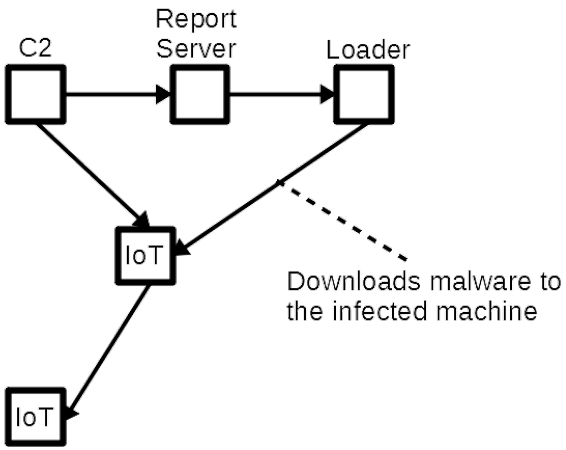
- The address and credentials of the victim machines where then sent to a report server
- This information could later be used by the Command and Control (C2) server

Spreading



- The records server then sent this information to the load program

Spreading



- The records server then sent this information to the load program
- This program would download a binary onto the victim and run the program

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary
- The machine would then run the binary and change the process information to make it harder to detect

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary
- The machine would then run the binary and change the process information to make it harder to detect
- The binary is then deleted.

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary
- The machine would then run the binary and change the process information to make it harder to detect
- The binary is then deleted.
  - This means infections won't carry across reboots

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary
- The machine would then run the binary and change the process information to make it harder to detect
- The binary is then deleted.
  - This means infections won't carry across reboots
- Once the victim is infected, it starts scanning

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary
- The machine would then run the binary and change the process information to make it harder to detect
- The binary is then deleted.
  - This means infections won't carry across reboots
- Once the victim is infected, it starts scanning
  - It would specifically avoid scanning servers owned by major corporations or the government

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary
- The machine would then run the binary and change the process information to make it harder to detect
- The binary is then deleted.
  - This means infections won't carry across reboots
- Once the victim is infected, it starts scanning
  - It would specifically avoid scanning servers owned by major corporations or the government
  - These entities would likely be too secure for this simple attack

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary
- The machine would then run the binary and change the process information to make it harder to detect
- The binary is then deleted.
  - This means infections won't carry across reboots
- Once the victim is infected, it starts scanning
  - It would specifically avoid scanning servers owned by major corporations or the government
  - These entities would likely be too secure for this simple attack
  - This also allowed the bot to keep a lower profile

Loading the Binary

- The aforementioned loader program downloaded an architecture specific binary
- The machine would then run the binary and change the process information to make it harder to detect
- The binary is then deleted.
  - This means infections won't carry across reboots
- Once the victim is infected, it starts scanning
  - It would specifically avoid scanning servers owned by major corporations or the government
  - These entities would likely be too secure for this simple attack
  - This also allowed the bot to keep a lower profile
  - These organizations would be much more likely to start search for and exploiting weaknesses in the malware if it infected their machines

Internet of Things Security

- The Internet of Things

Internet of Things Security

- The Internet of Things
  - Includes security cameras, routers, network-access storage, TV receivers, printer, DVRs, etc.

Internet of Things Security

- The Internet of Things
  - Includes security cameras, routers, network-access storage, TV receivers, printer, DVRs, etc.
  - Typically embedded systems that aren't powerful

Internet of Things Security

- The Internet of Things
  - Includes security cameras, routers, network-access storage, TV receivers, printer, DVRs, etc.
  - Typically embedded systems that aren't powerful
- Manufactures neglect security

Internet of Things Security

- The Internet of Things
  - Includes security cameras, routers, network-access storage, TV receivers, printer, DVRs, etc.
  - Typically embedded systems that aren't powerful
- Manufactures neglect security
  - Many manufactures use one user name and password

Internet of Things Security

- The Internet of Things
  - Includes security cameras, routers, network-access storage, TV receivers, printer, DVRs, etc.
  - Typically embedded systems that aren't powerful
- Manufactures neglect security
  - Many manufactures use one user name and password
  - Common passwords are frequent. `password`, `admin`, etc.

Internet of Things Security

- The Internet of Things
  - Includes security cameras, routers, network-access storage, TV receivers, printer, DVRs, etc.
  - Typically embedded systems that aren't powerful
- Manufactures neglect security
  - Many manufactures use one user name and password
  - Common passwords are frequent. `password`, `admin`, etc.
  - Some devices even have credentials hard coded in firmware

Internet of Things Security

- The Internet of Things
  - Includes security cameras, routers, network-access storage, TV receivers, printer, DVRs, etc.
  - Typically embedded systems that aren't powerful
- Manufactures neglect security
  - Many manufactures use one user name and password
  - Common passwords are frequent. `password`, `admin`, etc.
  - Some devices even have credentials hard coded in firmware
  - Most companies don't have the infrastructure to release patches for these systems

Internet of Things Security

- The Internet of Things
  - Includes security cameras, routers, network-access storage, TV receivers, printer, DVRs, etc.
  - Typically embedded systems that aren't powerful
- Manufactures neglect security
  - Many manufactures use one user name and password
  - Common passwords are frequent. `password`, `admin`, etc.
  - Some devices even have credentials hard coded in firmware
  - Most companies don't have the infrastructure to release patches for these systems
- This allowed the bot to easily infect a large number of machines

Disadvantages

- These less powerful devices also hurt Mirai's growth

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes
    - Compare to 37-minutes for Code-Red

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes
    - Compare to 37-minutes for Code-Red
    - 9-minutes for Blaster

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes
    - Compare to 37-minutes for Code-Red
    - 9-minutes for Blaster
  - Most bots scanned at less than 250 bytes per second

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes
    - Compare to 37-minutes for Code-Red
    - 9-minutes for Blaster
  - Most bots scanned at less than 250 bytes per second
  - Much slower than other bot nets

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes
    - Compare to 37-minutes for Code-Red
    - 9-minutes for Blaster
  - Most bots scanned at less than 250 bytes per second
  - Much slower than other bot nets
  - SQL Slammer was about 6000 times faster at 1.5 megabytes per second

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes
    - Compare to 37-minutes for Code-Red
    - 9-minutes for Blaster
  - Most bots scanned at less than 250 bytes per second
  - Much slower than other bot nets
  - SQL Slammer was about 6000 times faster at 1.5 megabytes per second
- Most devices were found in low bandwidth countries

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes
    - Compare to 37-minutes for Code-Red
    - 9-minutes for Blaster
  - Most bots scanned at less than 250 bytes per second
  - Much slower than other bot nets
  - SQL Slammer was about 6000 times faster at 1.5 megabytes per second
- Most devices were found in low bandwidth countries
  - Most infected devices were from South America and South-east Asia

Disadvantages

- These less powerful devices also hurt Mirai's growth
  - Mirai had a doubling time of 75-minutes
    - Compare to 37-minutes for Code-Red
    - 9-minutes for Blaster
  - Most bots scanned at less than 250 bytes per second
  - Much slower than other bot nets
  - SQL Slammer was about 6000 times faster at 1.5 megabytes per second
- Most devices were found in low bandwidth countries
  - Most infected devices were from South America and South-east Asia
  - Brazil, Colombia, and Vietnam hosted most of the bots

How DDoS Works

- How to DDoS for Dummies
  - A DDoS seeks to restrict a servers capabilities to respond to users by flooding it with requests from multiple different machines.

How DDoS Works

- How to DDoS for Dummies
  - A DDoS seeks to restrict a servers capabilities to respond to users by flooding it with requests from multiple different machines.
  - DDoS attacks are more difficult to protect against compared to a DOS attack because it is difficult to blacklist multiple IP Addresses and distinguish between real requests and the attack.

Volumetric Attacks

- Volumetric Attacks consist of a flooding a server with request packets to overwhelm its ability to respond. Volumetric attacks require little work to generate a high count of requests. With requests from multiple machines, it is difficult to prevent or dampen an attack on a server.

Protocal Attacks

- Protocol Attacks seek to disable a server by exloiting a weakness in a protocl. SYN flood attacks TCP by exploiting the three-way handshake process to create a backlogged queue. Ping attacks uses a large number of pings to attack a server. UDP floods send massive amounts of packets to random ports to overwhelm the queue of responses.

Application Layer Attacks

- Application layer attacks attempt to exploit the layer of human interaction with a machine. These attacks are nearly indistinguishable from real user interaction. They requires far less resources to execute this attack than it takes to prevent that attack. This makes these attacks resource efficient for an attacker.

- Most attacks were orchestrated against targets in the United States(50.3%), France(6.6%), and the UK(6.1%).

- Most attacks were orchestrated against targets in the United States(50.3%), France(6.6%), and the UK(6.1%).
- Mirai could also target particular ports to affect specific services. Of the attacks on specified ports, the most common ones attacked were 80(HTTP, 37.5%), 25565(Minecraft, 9.2%), 443(HTTPS, 6.4%), and 23594(Runescape, 3.4%).

- Most attacks were orchestrated against targets in the United States(50.3%), France(6.6%), and the UK(6.1%).
- Mirai could also target particular ports to affect specific services. Of the attacks on specified ports, the most common ones attacked were 80(HTTP, 37.5%), 25565(Minecraft, 9.2%), 443(HTTPS, 6.4%), and 23594(Runescape, 3.4%).
- Several Mirai C2 servers were attacked by some of its other C2 servers. These are from renting DDoS attackers against other renting DDoS attackers.

Attacks

- General Targets by Mirai

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)
  - Political Websites, anti-DDoS protectors

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)
  - Political Websites, anti-DDoS protectors
- Notable Targets

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)
  - Political Websites, anti-DDoS protectors
- Notable Targets
  - Krebs on Security

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)
  - Political Websites, anti-DDoS protectors
- Notable Targets
  - Krebs on Security
    - This was a high-profile attack on a well-known security blog

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)
  - Political Websites, anti-DDoS protectors
- Notable Targets
  - Krebs on Security
    - This was a high-profile attack on a well-known security blog
    - The attack peaked at around 600 Gbps, the largest Akamai had ever seen

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)
  - Political Websites, anti-DDoS protectors
- Notable Targets
  - Krebs on Security
    - This was a high-profile attack on a well-known security blog
    - The attack peaked at around 600 Gbps, the largest Akamai had ever seen
    - This forced them to drop Krebs as a client due to high costs

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)
  - Political Websites, anti-DDoS protectors
- Notable Targets
  - Krebs on Security
    - This was a high-profile attack on a well-known security blog
    - The attack peaked at around 600 Gbps, the largest Akamai had ever seen
    - This forced them to drop Krebs as a client due to high costs
  - Dyn - DNS attack disrupted access for Amazon, Github, Netflix, Twitter, and others

Attacks

- General Targets by Mirai
  - Multiple DDoS attacks against a variety of targets
  - Game Servers (primarily Minecraft and Runescape)
  - Political Websites, anti-DDoS protectors
- Notable Targets
  - Krebs on Security
    - This was a high-profile attack on a well-known security blog
    - The attack peaked at around 600 Gbps, the largest Akamai had ever seen
    - This forced them to drop Krebs as a client due to high costs
  - Dyn - DNS attack disrupted access for Amazon, Github, Netflix, Twitter, and others
  - Lonestar Cell - most attacked target, destroyed internet capabilities in Liberia

Network Telescope

- Purpose: to analyze the growth and size of the botnet

Network Telescope

- Purpose: to analyze the growth and size of the botnet
- Monitored network request (scan) to a network telescope composed 4.7 millions IP address.

Network Telescope

- Purpose: to analyze the growth and size of the botnet
- Monitored network request (scan) to a network telescope composed 4.7 millions IP address.
  - On average, the network telescope received 1.1 million packets from 269,000 IP addresses per minute

Network Telescope

- Purpose: to analyze the growth and size of the botnet
- Monitored network request (scan) to a network telescope composed 4.7 millions IP address.
  - On average, the network telescope received 1.1 million packets from 269,000 IP addresses per minute
  - Observed 116.2 billions Mirai probes from 55.4 millions IP address.

Network Telescope

- Purpose: to analyze the growth and size of the botnet
- Monitored network request (scan) to a network telescope composed 4.7 millions IP address.
  - On average, the network telescope received 1.1 million packets from 269,000 IP addresses per minute
  - Observed 116.2 billions Mirai probes from 55.4 millions IP address.
- A raw count of IP addess is a poor metric due to DHCP churn

Network Telescope

- Purpose: to analyze the growth and size of the botnet
- Monitored network request (scan) to a network telescope composed 4.7 millions IP address.
  - On average, the network telescope received 1.1 million packets from 269,000 IP addresses per minute
  - Observed 116.2 billions Mirai probes from 55.4 millions IP address.
- A raw count of IP addess is a poor metric due to DHCP churn
  - Consider the number of hosts actively "scanning" at the start of every hours.

Network Telescope

- Purpose: to analyze the growth and size of the botnet
- Monitored network request (scan) to a network telescope composed 4.7 millions IP address.
  - On average, the network telescope received 1.1 million packets from 269,000 IP addresses per minute
  - Observed 116.2 billions Mirai probes from 55.4 millions IP address.
- A raw count of IP addess is a poor metric due to DHCP churn
  - Consider the number of hosts actively "scanning" at the start of every hours.
  - Identified scans that targeted the IPv4 address space at an estimated rate of at least five packets per second.

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).
- Focus on scans of HTTPS, FTP, SSH, Telnet, and CWMP.

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).
- Focus on scans of HTTPS, FTP, SSH, Telnet, and CWMP.
- Difficulties and challenges to make accurate device labeling:

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).
- Focus on scans of HTTPS, FTP, SSH, Telnet, and CWMP.
- Difficulties and challenges to make accurate device labeling:
  - Mirai prevents infected devices from being scanned upon infection.

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).
- Focus on scans of HTTPS, FTP, SSH, Telnet, and CWMP.
- Difficulties and challenges to make accurate device labeling:
  - Mirai prevents infected devices from being scanned upon infection.
  - The scan often takes 24 hours to complete, during which devices may churn to a new IP address

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).
- Focus on scans of HTTPS, FTP, SSH, Telnet, and CWMP.
- Difficulties and challenges to make accurate device labeling:
  - Mirai prevents infected devices from being scanned upon infection.
  - The scan often takes 24 hours to complete, during which devices may churn to a new IP address
  - Resolution: restricting analysis to banners that were collected within twenty minutes of scanning activity.

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).
- Focus on scans of HTTPS, FTP, SSH, Telnet, and CWMP.
- Difficulties and challenges to make accurate device labeling:
  - Mirai prevents infected devices from being scanned upon infection.
  - The scan often takes 24 hours to complete, during which devices may churn to a new IP address
  - Resolution: restricting analysis to banners that were collected within twenty minutes of scanning activity.
- Post-filtering, the dataset include 1.8 millions banner associated with 1.2 million IP address.

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).
- Focus on scans of HTTPS, FTP, SSH, Telnet, and CWMP.
- Difficulties and challenges to make accurate device labeling:
  - Mirai prevents infected devices from being scanned upon infection.
  - The scan often takes 24 hours to complete, during which devices may churn to a new IP address
  - Resolution: restricting analysis to banners that were collected within twenty minutes of scanning activity.
- Post-filtering, the dataset include 1.8 millions banner associated with 1.2 million IP address.
  - Process each banner to identify the device manufacturer and model using Nmap.

Active Scanning

- Purpose: to analyze infected device composition (manufacturer and model).
- Focus on scans of HTTPS, FTP, SSH, Telnet, and CWMP.
- Difficulties and challenges to make accurate device labeling:
  - Mirai prevents infected devices from being scanned upon infection.
  - The scan often takes 24 hours to complete, during which devices may churn to a new IP address
  - Resolution: restricting analysis to banners that were collected within twenty minutes of scanning activity.
- Post-filtering, the dataset include 1.8 millions banner associated with 1.2 million IP address.
  - Process each banner to identify the device manufacturer and model using Nmap.
  - In total, identified for 31.5 % of banners (about 600k banners)

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.
  - The honeypot logged all incoming traffic and downloaded any binaries that the attackers attemps to install.

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.
  - The honeypot logged all incoming traffic and downloaded any binaries that the attackers attemps to install.
  - Block all outgoing request to avoid collateral damages.

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.
  - The honeypot logged all incoming traffic and downloaded any binaries that the attackers attemps to install.
  - Block all outgoing request to avoid collateral damages.
- Logged 80K connection attempts from 54K IP addresses and collected 141 unique binaries.

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.
  - The honeypot logged all incoming traffic and downloaded any binaries that the attackers attemps to install.
  - Block all outgoing request to avoid collateral damages.
- Logged 80K connection attempts from 54K IP addresses and collected 141 unique binaries.
  - Supplemented these data with unique binaries from others

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.
  - The honeypot logged all incoming traffic and downloaded any binaries that the attackers attemps to install.
  - Block all outgoing request to avoid collateral damages.
- Logged 80K connection attempts from 54K IP addresses and collected 141 unique binaries.
  - Supplemented these data with unique binaries from others
  - In totals, they collected 1028 unique binaries.

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.
  - The honeypot logged all incoming traffic and downloaded any binaries that the attackers attemps to install.
  - Block all outgoing request to avoid collateral damages.
- Logged 80K connection attempts from 54K IP addresses and collected 141 unique binaries.
  - Supplemented these data with unique binaries from others
  - In totals, they collected 1028 unique binaries.
- Analyzed binaries for MIPS 32-bit, ARM 32-bit, and x86 32-bit.

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.
  - The honeypot logged all incoming traffic and downloaded any binaries that the attackers attemps to install.
  - Block all outgoing request to avoid collateral damages.
- Logged 80K connection attempts from 54K IP addresses and collected 141 unique binaries.
  - Supplemented these data with unique binaries from others
  - In totals, they collected 1028 unique binaries.
- Analyzed binaries for MIPS 32-bit, ARM 32-bit, and x86 32-bit.
  - Extracted the set of logins, password, IP blacklisted, C2 domains.

Telnet Honeypots

- Use a set of Telnet honeypots that masqueraded as vulnerable IoT.
  - The honeypot logged all incoming traffic and downloaded any binaries that the attackers attemps to install.
  - Block all outgoing request to avoid collateral damages.
- Logged 80K connection attempts from 54K IP addresses and collected 141 unique binaries.
  - Supplemented these data with unique binaries from others
  - In totals, they collected 1028 unique binaries.
- Analyzed binaries for MIPS 32-bit, ARM 32-bit, and x86 32-bit.
  - Extracted the set of logins, password, IP blacklisted, C2 domains.
  - Identified 67 C2 domains and 48 distinct username password dictionaries (containing a total 371 unique passwords)

Active & Passive DNS

- Purpose: to construct a graph reflecting the shared infrastructure used by Mirai.

Active & Passive DNS

- Purpose: to construct a graph reflecting the shared infrastructure used by Mirai.
- Collected 209 millions passive DNS record per day (historical record of DNS zone) and 290 millions active DNS record per day of C2 server.

Active & Passive DNS

- Purpose: to construct a graph reflecting the shared infrastructure used by Mirai.
- Collected 209 millions passive DNS record per day (historical record of DNS zone) and 290 millions active DNS record per day of C2 server.
- Use above dataset to identify shared DNS infrastructure by linking related historic domain names (RHDN) and related historic IPs (RHIPs).

Active & Passive DNS

- Purpose: to construct a graph reflecting the shared infrastructure used by Mirai.
- Collected 209 millions passive DNS record per day (historical record of DNS zone) and 290 millions active DNS record per day of C2 server.
- Use above dataset to identify shared DNS infrastructure by linking related historic domain names (RHDN) and related historic IPs (RHIPs).
  - For a given C2 domain, identify the IP address it previously resolved to and added them to a growing set of domains and IPs.

Active & Passive DNS

- Purpose: to construct a graph reflecting the shared infrastructure used by Mirai.
- Collected 209 millions passive DNS record per day (historical record of DNS zone) and 290 millions active DNS record per day of C2 server.
- Use above dataset to identify shared DNS infrastructure by linking related historic domain names (RHDN) and related historic IPs (RHIPs).
  - For a given C2 domain, identify the IP address it previously resolved to and added them to a growing set of domains and IPs.
  - Starting from an IP and finding any domain names that concurrently resolved it.

Active & Passive DNS

- Purpose: to construct a graph reflecting the shared infrastructure used by Mirai.
- Collected 209 millions passive DNS record per day (historical record of DNS zone) and 290 millions active DNS record per day of C2 server.
- Use above dataset to identify shared DNS infrastructure by linking related historic domain names (RHDN) and related historic IPs (RHIPs).
  - For a given C2 domain, identify the IP address it previously resolved to and added them to a growing set of domains and IPs.
  - Starting from an IP and finding any domain names that concurrently resolved it.
  - In the end, from a single domain name, we can expand a set of domain name and IP addresses.

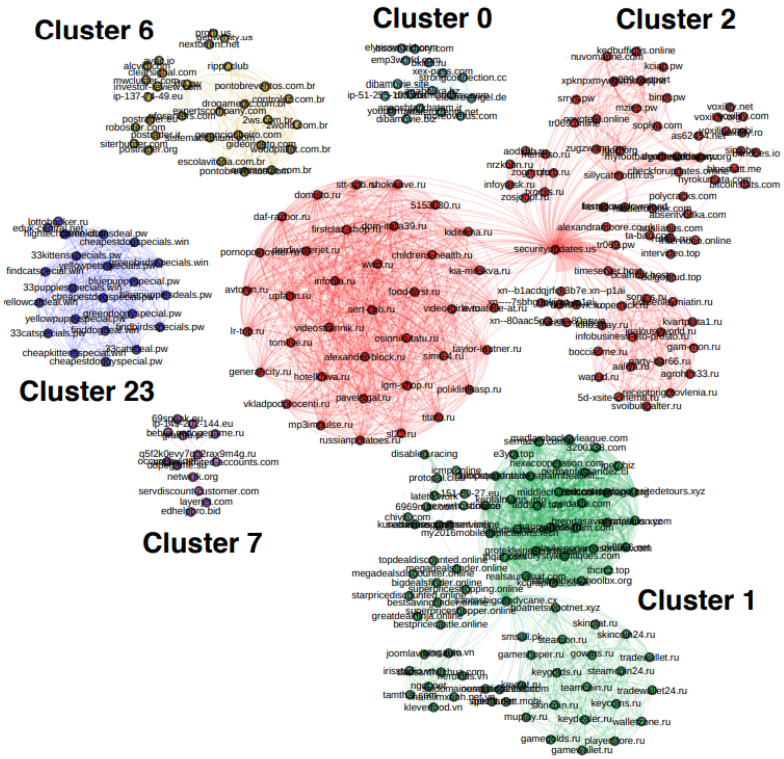


Figure: C2 Domain Relationships – We visualize related C2 infrastructure, depicting C2 domains as nodes and shared IPs as edges between two domains.

Attack Commands

- Purpose: to track the attack commands issued by the Mirai operators.

Attack Commands

- Purpose: to track the attack commands issued by the Mirai operators.
- Simulated a Mirai-infected device and communicated with the C2 server using a custom bot-to-C2 protocol .

Attack Commands

- Purpose: to track the attack commands issued by the Mirai operators.
- Simulated a Mirai-infected device and communicated with the C2 server using a custom bot-to-C2 protocol .
- In total, Akamai observed 64K attack commands issued by 484 unique C2 servers (by IP address).

Attack Commands

- Purpose: to track the attack commands issued by the Mirai operators.
- Simulated a Mirai-infected device and communicated with the C2 server using a custom bot-to-C2 protocol .
- In total, Akamai observed 64K attack commands issued by 484 unique C2 servers (by IP address).
  - This is a naive analysis because individual C2 servers often repeat the same attack command in rapid succession.

Attack Commands

- Purpose: to track the attack commands issued by the Mirai operators.
- Simulated a Mirai-infected device and communicated with the C2 server using a custom bot-to-C2 protocol .
- In total, Akamai observed 64K attack commands issued by 484 unique C2 servers (by IP address).
  - This is a naive analysis because individual C2 servers often repeat the same attack command in rapid succession.
  - Resolution: collapse matching commands that occur within 90 seconds of each others.

Attack Commands

- Purpose: to track the attack commands issued by the Mirai operators.
- Simulated a Mirai-infected device and communicated with the C2 server using a custom bot-to-C2 protocol .
- In total, Akamai observed 64K attack commands issued by 484 unique C2 servers (by IP address).
  - This is a naive analysis because individual C2 servers often repeat the same attack command in rapid succession.
  - Resolution: collapse matching commands that occur within 90 seconds of each others.
  - Results: 15,194 attacks from 146 unique IP clusters, which cover the Dyn attack and Liberia attacks.

DDoS Attack trace

- Purpose: to corroborate the IP addresses observed in attacks versus those found scanning our network telescope.

DDoS Attack trace

- Purpose: to corroborate the IP addresses observed in attacks versus those found scanning our network telescope.
  - Dyn provided 107.5K IP addresses associated with attack on October 21, 2016 and 158.8K IP addresses involved in attack on September 25, 2016.

DDoS Attack trace

- Purpose: to corroborate the IP addresses observed in attacks versus those found scanning our network telescope.
  - Dyn provided 107.5K IP addresses associated with attack on October 21, 2016 and 158.8K IP addresses involved in attack on September 25, 2016.
  - Form a statistics to calculate what fraction of these IP addresses matched the list of IP address obseved by our network telescope.

Releasing the Source Code

- The source code was released on September 30, 2016

Releasing the Source Code

- The source code was released on September 30, 2016
- A user named “Anna-senpai” released the source code for free on hackerforums.net

Releasing the Source Code

- The source code was released on September 30, 2016
- A user named "Anna-senpai" released the source code for free on hackerforums.net
- This spawned a number of copycat attacks with variations on the original bot

Releasing the Source Code

- The source code was released on September 30, 2016
- A user named "Anna-senpai" released the source code for free on hackerforums.net
- This spawned a number of copycat attacks with variations on the original bot
  - Many included new exploits, modified dictionaries, and different IP blacklists

Releasing the Source Code

- The source code was released on September 30, 2016
- A user named "Anna-senpai" released the source code for free on hackerforums.net
- This spawned a number of copycat attacks with variations on the original bot
  - Many included new exploits, modified dictionaries, and different IP blacklists
- These botnets eventually starting competing

Releasing the Source Code

- The source code was released on September 30, 2016
- A user named "Anna-senpai" released the source code for free on hackerforums.net
- This spawned a number of copycat attacks with variations on the original bot
  - Many included new exploits, modified dictionaries, and different IP blacklists
- These botnets eventually starting competing
  - Killing processes started by similar bots

Releasing the Source Code

- The source code was released on September 30, 2016
- A user named "Anna-senpai" released the source code for free on hackerforums.net
- This spawned a number of copycat attacks with variations on the original bot
  - Many included new exploits, modified dictionaries, and different IP blacklists
- These botnets eventually starting competing
  - Killing processes started by similar bots
  - Closing the ports used to attack the machine

Releasing the Source Code

- The source code was released on September 30, 2016
- A user named "Anna-senpai" released the source code for free on hackerforums.net
- This spawned a number of copycat attacks with variations on the original bot
  - Many included new exploits, modified dictionaries, and different IP blacklists
- These botnets eventually starting competing
  - Killing processes started by similar bots
  - Closing the ports used to attack the machine
  - At various points, competing command and control servers were subject to DDoS attacks

Defense Against the Dark Arts

- Industry Improvements

Defense Against the Dark Arts

- Industry Improvements
  - Randomized default passwords

Defense Against the Dark Arts

- Industry Improvements
  - Randomized default passwords
  - Closed ports on default

Defense Against the Dark Arts

- Industry Improvements
  - Randomized default passwords
  - Closed ports on default
  - Employ Automatic Updates and Self-Monitoring

Defense Against the Dark Arts

- Industry Improvements
  - Randomized default passwords
  - Closed ports on default
  - Employ Automatic Updates and Self-Monitoring
  - Creating Standard for Model and Firmware identification

Defense Against the Dark Arts

- Industry Improvements
  - Randomized default passwords
  - Closed ports on default
  - Employ Automatic Updates and Self-Monitoring
  - Creating Standard for Model and Firmware identification
- User Improvements

Defense Against the Dark Arts

- Industry Improvements
  - Randomized default passwords
  - Closed ports on default
  - Employ Automatic Updates and Self-Monitoring
  - Creating Standard for Model and Firmware identification
- User Improvements
  - Create good secure credentials and not use the defaults

Defense Against the Dark Arts

- Industry Improvements
  - Randomized default passwords
  - Closed ports on default
  - Employ Automatic Updates and Self-Monitoring
  - Creating Standard for Model and Firmware identification
- User Improvements
  - Create good secure credentials and not use the defaults
  - Purchase from reputable and secure companies

Defense Against the Dark Arts

- Industry Improvements
  - Randomized default passwords
  - Closed ports on default
  - Employ Automatic Updates and Self-Monitoring
  - Creating Standard for Model and Firmware identification
- User Improvements
  - Create good secure credentials and not use the defaults
  - Purchase from reputable and secure companies
  - Replace old and unsupported devices

Defense Against the Dark Arts

- Randomized default passwords prevent attackers from employing a dictionary of default passwords.
- Having ports not used default to closed mitigates the chances of a successful attack.
- Automatic updates prevent users from refusing updates during hours of use and keeps systems secure against previous exploits. Bug bounties encourage the community to find and report all possible exploits to be patched.
- Standards for model and version identification allow server admins to easily see any and all machines that have known vulnerabilities.

Defense Against the Dark Arts

- Users should create secure usernames and passwords for all devices to mitigate the chance of it being hacked using brute force.
- Smart purchases from known and trusted companies that prioritize security of their manufactured devices acts as a deterrent from would be attackers.
- Old and unsupported devices should be replaced with newer models that conform with current security standards and have strong customer support.

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha
  - 20-year-old Josiah White

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha
  - 20-year-old Josiah White
  - 21-year-old Dalton Norman

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha
  - 20-year-old Josiah White
  - 21-year-old Dalton Norman
- It's likely named after the anime Mirai Nikki (Future Diary)

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha
  - 20-year-old Josiah White
  - 21-year-old Dalton Norman
- It's likely named after the anime Mirai Nikki (Future Diary)
- As revealed after the trial, this botnet was originally created to DDoS Minecraft servers

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha
  - 20-year-old Josiah White
  - 21-year-old Dalton Norman
- It's likely named after the anime Mirai Nikki (Future Diary)
- As revealed after the trial, this botnet was originally created to DDoS Minecraft servers
- The group allegedly started trying to profit from their botnet

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha
  - 20-year-old Josiah White
  - 21-year-old Dalton Norman
- It's likely named after the anime Mirai Nikki (Future Diary)
- As revealed after the trial, this botnet was originally created to DDoS Minecraft servers
- The group allegedly started trying to profit from their botnet
  - Extorting profitable servers

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha
  - 20-year-old Josiah White
  - 21-year-old Dalton Norman
- It's likely named after the anime Mirai Nikki (Future Diary)
- As revealed after the trial, this botnet was originally created to DDoS Minecraft servers
- The group allegedly started trying to profit from their botnet
  - Extorting profitable servers
  - Creating a company to protect server owners from DDoS attacks

Origin of Mirai

- After this paper was released, we started to learn more about the botnet's origins
- Mirai was created by 3 computer science students
  - 21-year-old Paras Jha
  - 20-year-old Josiah White
  - 21-year-old Dalton Norman
- It's likely named after the anime Mirai Nikki (Future Diary)
- As revealed after the trial, this botnet was originally created to DDoS Minecraft servers
- The group allegedly started trying to profit from their botnet
  - Extorting profitable servers
  - Creating a company to protect server owners from DDoS attacks
  - Renting out their botnet to other cybercriminals

Investigations

- Anna-senpai boosted about his programming skills on the forum

Investigations

- Anna-senpai boosted about his programming skills on the forum
  - After investigating companies related to DataWagon, Krebs found ProTraf solutions

Investigations

- Anna-senpai boosted about his programming skills on the forum
  - After investigating companies related to DataWagon, Krebs found ProTraf solutions
  - The skills Anna-senpai had were extremely similar to Jha's.

Investigations

- Anna-senpai boosted about his programming skills on the forum
  - After investigating companies related to DataWagon, Krebs found ProTraf solutions
  - The skills Anna-senpai had were extremely similar to Jha's.
  - He also has experience running Minecraft servers, similar to the original targets

Investigations

- Anna-senpai boosted about his programming skills on the forum
  - After investigating companies related to DataWagon, Krebs found ProTraf solutions
  - The skills Anna-senpai had were extremely similar to Jha's.
  - He also has experience running Minecraft servers, similar to the original targets
- According to Krebs on Security, the creators then released the source code to "distance themselves from their creation"

Investigations

- Anna-senpai boosted about his programming skills on the forum
  - After investigating companies related to DataWagon, Krebs found ProTraf solutions
  - The skills Anna-senpai had were extremely similar to Jha's.
  - He also has experience running Minecraft servers, similar to the original targets
- According to Krebs on Security, the creators then released the source code to "distance themselves from their creation"
- Many attacks, such as the attack on Dyn, are believed to be a result of copy cat attackers

Investigation

- On December 8, 2017, the FBI announced that it had secured guilty pleas

Investigation

- On December 8, 2017, the FBI announced that it had secured guilty pleas
- The authors were able to avoid jail time after cooperating with the FBI to help track other cybercriminals.

Investigation

- On December 8, 2017, the FBI announced that it had secured guilty pleas
- The authors were able to avoid jail time after cooperating with the FBI to help track other cybercriminals.
- This allowed them to avoid jail time

Investigation

- On December 8, 2017, the FBI announced that it had secured guilty pleas
- The authors were able to avoid jail time after cooperating with the FBI to help track other cybercriminals.
- This allowed them to avoid jail time
  - Five-year probation

Investigation

- On December 8, 2017, the FBI announced that it had secured guilty pleas
- The authors were able to avoid jail time after cooperating with the FBI to help track other cybercriminals.
- This allowed them to avoid jail time
  - Five-year probation
  - 2,500 Hours of community service

Investigation

- On December 8, 2017, the FBI announced that it had secured guilty pleas
- The authors were able to avoid jail time after cooperating with the FBI to help track other cybercriminals.
- This allowed them to avoid jail time
  - Five-year probation
  - 2,500 Hours of community service
  - $127,000 in restitution

Investigation

- On December 8, 2017, the FBI announced that it had secured guilty pleas
- The authors were able to avoid jail time after cooperating with the FBI to help track other cybercriminals.
- This allowed them to avoid jail time
  - Five-year probation
  - 2,500 Hours of community service
  - $127,000 in restitution
  - They had to forfeit the concurrency made during that time

Investigation

- On December 8, 2017, the FBI announced that it had secured guilty pleas
- The authors were able to avoid jail time after cooperating with the FBI to help track other cybercriminals.
- This allowed them to avoid jail time
  - Five-year probation
  - 2,500 Hours of community service
  - $127,000 in restitution
  - They had to forfeit the concurrency made during that time
  - This was about 17 bitcoin

Conclusion

- Mirai was a centralized botnet that was responsible for some of the largest DoS attacks every recorded.

Conclusion

- Mirai was a centralized botnet that was responsible for some of the largest DoS attacks every recorded.
- This bot was relatively simple with small dictionaries

Conclusion

- Mirai was a centralized botnet that was responsible for some of the largest DoS attacks every recorded.
- This bot was relatively simple with small dictionaries
- They creators of this bot exploited the security negligence of hardware manufactures

Conclusion

- Mirai was a centralized botnet that was responsible for some of the largest DoS attacks every recorded.
- This bot was relatively simple with small dictionaries
- They creators of this bot exploited the security negligence of hardware manufactures
- They were able to quickly take over a large number of IoT devices

Conclusion

- Mirai was a centralized botnet that was responsible for some of the largest DoS attacks every recorded.
- This bot was relatively simple with small dictionaries
- They creators of this bot exploited the security negligence of hardware manufactures
- They were able to quickly take over a large number of IoT devices
- This attack served as a wake up call, prompting reform in these industries