The evolution of DDoS attacks and IoT Botnets

DoS Attacks

A denial of service attack (DOS) is any type of attack on a networking structure to disable a server from servicing its clients. Attacks range from sending millions of requests to a server in an attempt to slow it down, flooding a server with large packets of invalid data, to sending requests with an invalid or spoofed IP address

Business Losses

It is estimated that DDoS attacks cause an average billion-dollar loss every year

A little bit of History

DoS Attack’s Origins

The first DoS attack was done by 13-year-old David Dennis in 1974. Dennis wrote a program that forced some computers at a nearby university research lab to power off.

Evolving to DDoS

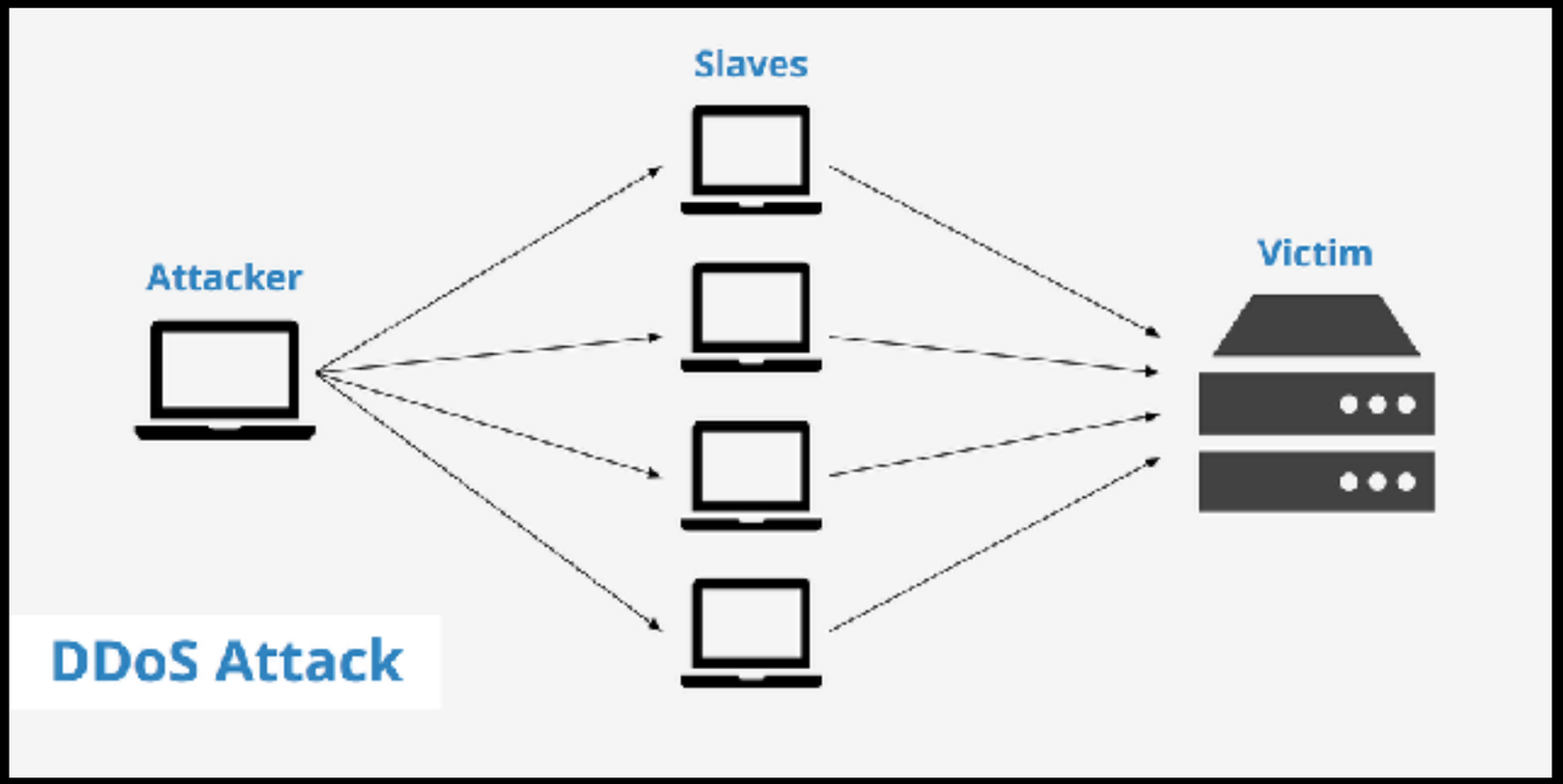
The first major DDoS attack to gain international attention was early in 2000, launched by a 15-year-old from Canada who called himself Mafiaboy. His campaign effectively broke the internet, restricting access to the web's most popular sites for a full week, including Yahoo!, Fifa.com, Amazon.com, eBay, CNN, Dell, and more.

Evolving to DDoS

Authorities said Mafiaboy broke into computers on more than 50 networks around the world - many operated by universities - in order to install and launch automated DDoS software known as Sinkhole. Mafiaboy then directed that remote-controlled software to bombard the targeted Web servers with so many requests they often buckled under the load.

The many charges against Mafiaboy stemmed largely from the individual break-ins at computers commandeered as Sinkhole "zombies."

Evolving to DDoS



Evolving to DDoS

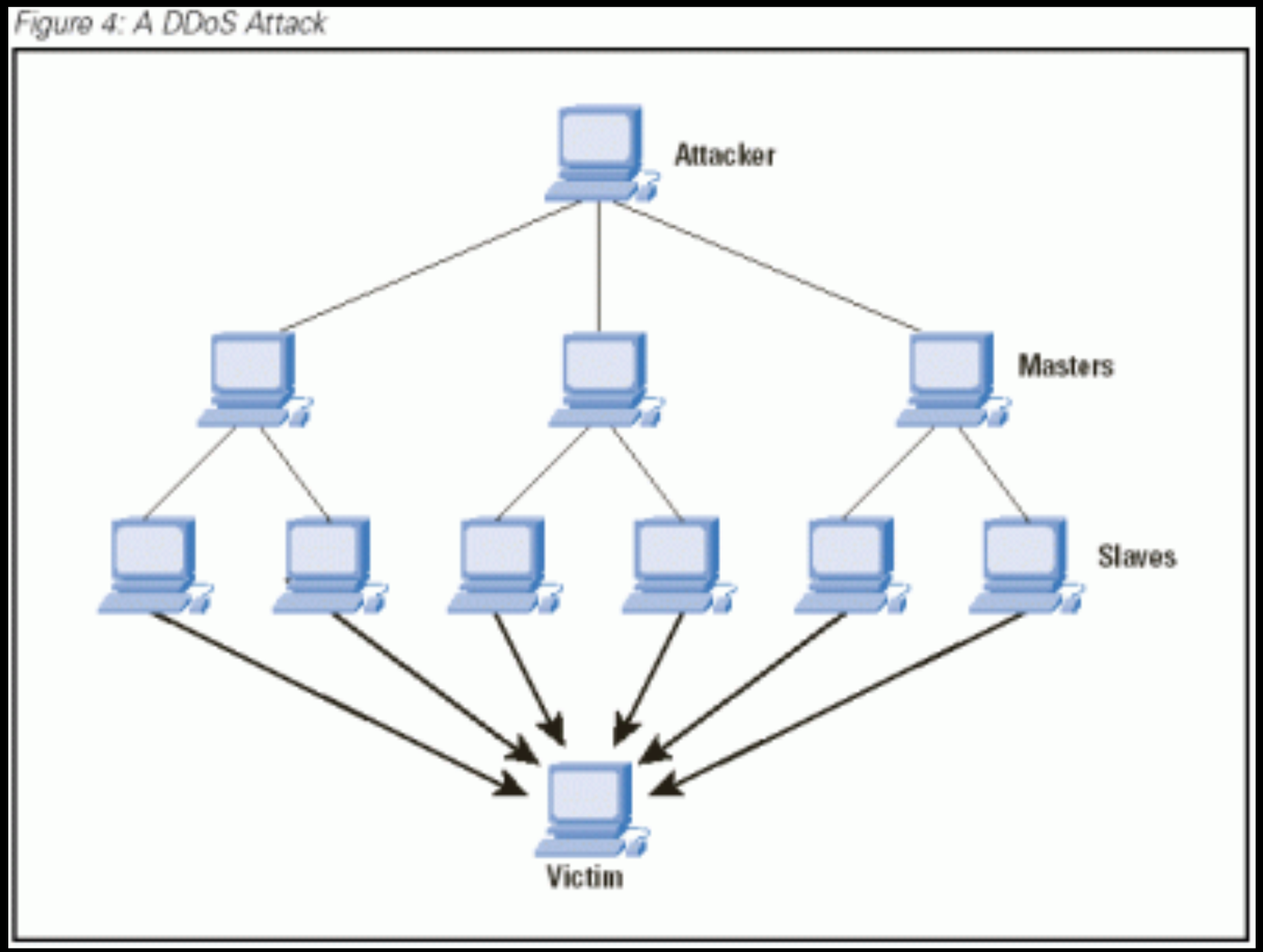
One of the first large-scale DDoS attacks occurred in August 1999, when a hacker used a tool called “Trinoo” to disable the University of Minnesota’s computer network for more than two days. Trinoo consisted of a network of compromised machines called “Masters” and “Daemons,” allowing an attacker to send a DoS instruction to a few Masters, which then forwarded instructions to the hundreds of Daemons to commence a UDP flood against the target IP address. The tool made no effort to hide the Daemons’ IP addresses, so the owners of the attacking systems were contacted and had no idea that their systems had been compromised and were being used in a DDoS attack.

Trinoo

Trinoo (also known as trin00) was the first well known DDos attack used against the University of Minnesota in August 1999. This two day attack involved flooding servers with UDP packets originating from thousands of machines. Source addresses were not spoofed, so systems running the offending daemons were contacted. However, the attacker responded simply by introducing new daemon machines into the attack. Trinoo was first found as a binary daemon on a number of compromised Solaris 2.x systems. Malicious code had been introduced through exploitation of buffer over-run bugs in the remote procedure call (RPC) services ‘statd’, ‘cmsd’ and ‘ttdbserverd’.

Trinoo

Figure 4: A DDoS Attack



Trinoo

Attackers can send a number of commands to masters.
Examples are:

- quit - to logoff from master
- Dos IP - to launch a DDoS attack against the adress IP
- mdos - to launch a multiple DDoS attack
- bcast - to form a list of started daemons

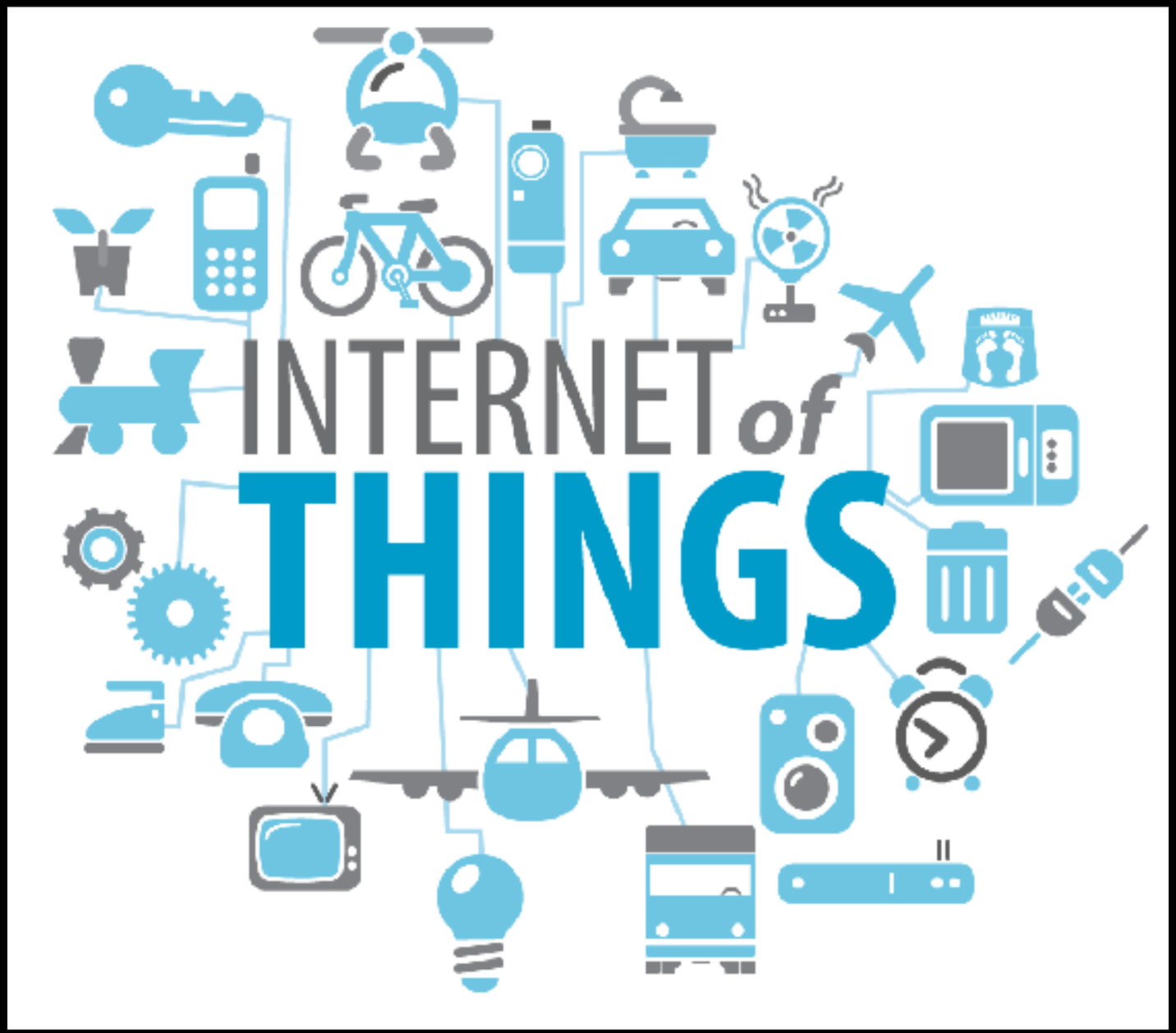
Trinoo

The trinoo DDos formulation begins with the attacker compromising one of many master systems. These systems are set-up with vulnerability scanning tools, root kits (to conceal malicious programs, files and connections), the master and trinoo daemon programs, and a list of vulnerable hosts (which are potential daemon systems). DDos attack preparation involves the master(s) scanning for systems exhibiting the vulnerabilities described above (typically Solaris 2.x and Linux systems). A list of vulnerable systems is then passed to an exploit script that compromises each system, sets up and connects a listening shell (tcp port 1524), and compiles a list of successful compromises – or 'owned' systems. The list of 'owned' systems is passed to another script that installs the trinoo daemon and a root kit via the open tcp port 1524 – completing the construction of the 'trinoo network'. (David Dittrich, 1999).

Trinoo

The DDos attack begins when the attacker connects (to masters) via telnet to tcp port 27665 and enters a password (the password was "betaalmostdone" in the case examined by Dittrich). Masters then pass command lines to daemons via UDP port 27444. These commands are password protected and are of the form: arg1 password arg2. Daemons respond to masters on UDP port 31335. Masters form a list of alive daemons by listening for the text "*HELLO*" in the data portion of UDP packets originating from daemons.

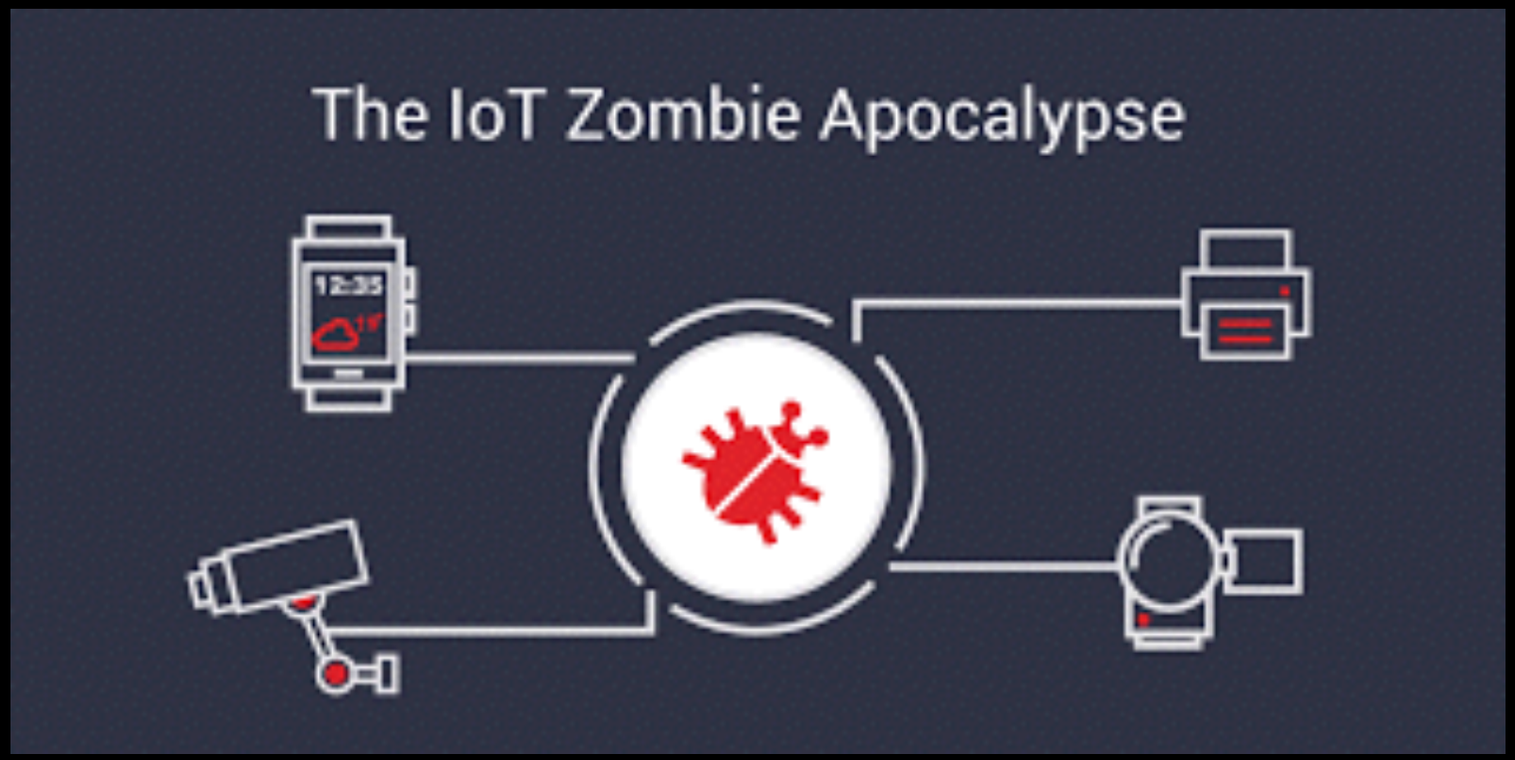
Rise of IoT



Top 8 IoT Vulnerabilities

1-Weak, Guessable, or Hardcoded Passwords.
2-Insecure Network Services.
3-Insecure Ecosystem Interfaces.
4-Lack of Secure Update Mechanism.
5-Use of Insecure or Outdated Components.
6-Insufficient Privacy Protection.
7-Insecure Data Transfer and Storage.
8-Lack of Device Management.

IoT Botnets



Most notorious IoT botnets

Bashlite

- Originally created in 2014
- Took advantage of the ShellShock vulnerability
- Infected mostly devices running BusyBox
- Mostly infected IoT devices

Mirai

- First discovered in 2016
- Seen by many as an evolution of Bashlite
- Infects mostly IoT devices

Bashlite

- Command and control servers (C&C) are the operators' interface to the botnet. C&Cs receive commands from operators and maintain connections with infected devices to broadcast commands.

- Bots are infected devices that are part of the botnet. Bots report their state to C&Cs and execute the received commands.

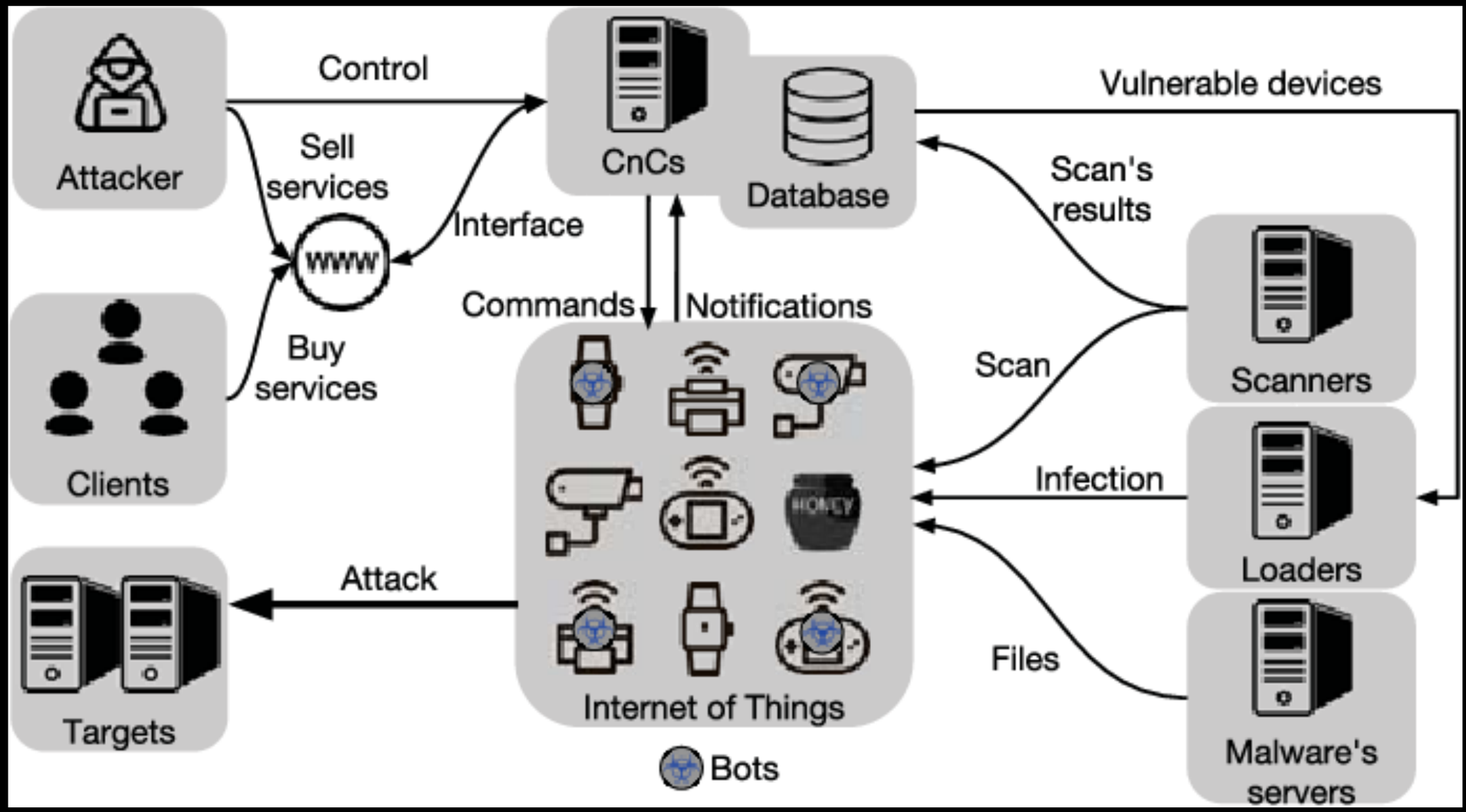
- Scanners probe devices to find telnet and SSH servers to attempt login and identify vulnerable devices.

- Loaders login to vulnerable devices to download and run the botnet malware, creating a new bot.

- Malware servers host resources used by the botnet such as shell scripts and executable binaries.

- Database (potentially distributed) stores information collected by the botnet, *e.g.*, active bots and scan results.

Bashlite


Attacker
Control
Sell
services
Interface
CnCs
Database
Vulnerable devices
Scan's
results
Scanners
Buy
services
Commands
Notifications
Scan
Clients
Infection
Loaders
Attack
Files
Targets
Internet of Things
Malware's
servers
Bots

Bashlite Commands

- Attack (66.4%) commands start DDoS attacks against select targets: !* TCPFLOOD 192.168.0.1 80 120 32 syn
- Management (18.4%) commands, *e.g.*, to update the malware binary, remove bots from the botnet, or enable scanning: !* UPDATE, !* BOTKILL, or !* SCAN ON
- Queries (1.27%) retrieve information about the botnet and its state: !* HELP and !* STATUS
- Interrupt (13.1%) commands stop ongoing attacks, if any: !* KILLATTK
- Other (0.70%) commands that do not fit in any of the previous classes: !* CLEAR

Bashlite Attacks

- Volumetric attacks correspond to 73.4%
- TCP-related attacks correspond to 13.6%
- Application attacks correspond to 13%

Bashlite Targets

- CDN / hosting corresponds to 14.84% of targets
- Transit / Access correspond to 82.20% of targets
- Enterprises correspond to 1.75% of targets

Mirai Improvements

- Bots themselves can perform vulnerability scans
- C&C are resolved using DNS instead of having the IPs hardcoded
- Commands are transmitted in binary form instead of plain text

Mirai Attacks

- Volumetric attacks correspond to 30%
- TCP-related attacks correspond to 38.4%
- Application attacks correspond to 30.6%

Mirai Targets

- CDN / hosting corresponds to 26.80% of targets
- Transit / Access correspond to 67.94% of targets
- Enterprises correspond to 3.92% of targets

References

- https://www.information-age.com/evolution-of-ddos-123473947/
- https://www.securityfocus.com/news/250
- A. Marzano et al., “The Evolution of Bashlite and Mirai IoT Botnets,” 2018 IEEE Symp. Comput. Commun., pp. 813–818, 2018.
- https://security.radware.com/ddos-knowledge-center/ddos-chronicles/ddos-attacks-history/
- https://www.giac.org/paper/gsec/28/distributed-denial-service-attack-tools-trinoo-wintrinoo/100225
- https://www.computerweekly.com/microscope/opinion/The-rise-of-IoT
- Ali, Bako & Awad, Ali. (2018). Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes. Sensors. 18. 817. 10.3390/s18030817.