NOKIA

The Coming of Age of IoT Botnets

Nokia Threat Intelligence Center

NOKIA

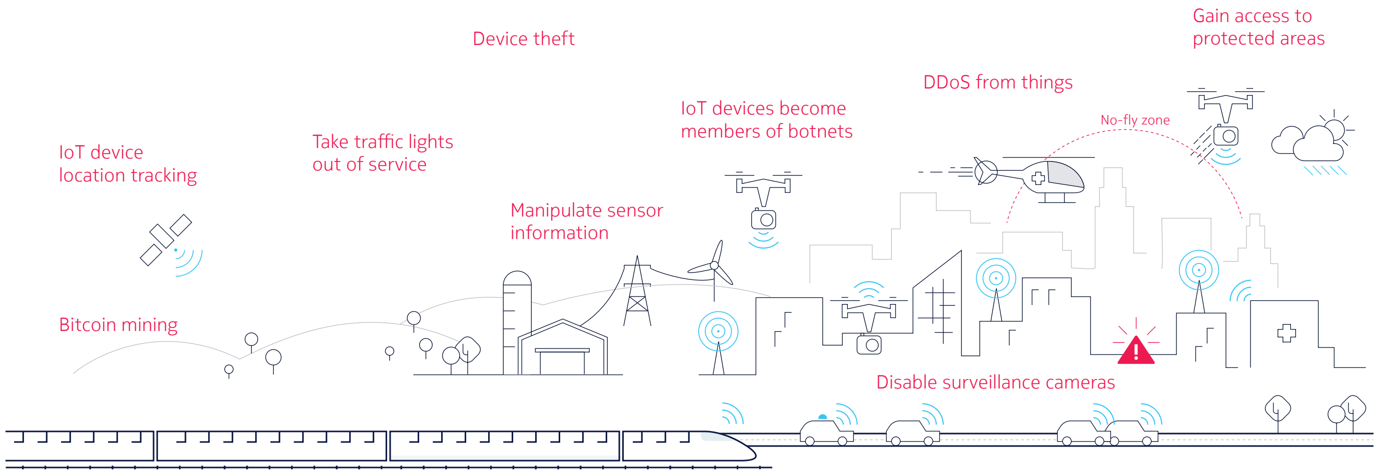
Contents

IoT Ecosystems

The concept of Internet of Things (IoT) is usually associated with physical devices that have the highest visibility, such as cameras, smart home automation or smart appliances. However, IoT comprises a system of interdependent components that are technologically very diverse and located in various locations.

Attacks on IoT systems are also very diverse, as illustrated in figure 1:

Figure 1. Diversity of attacks on IoT systems


Device theft
Gain access to protected areas
DDoS from things
IoT devices become members of botnets
No-fly zone
IoT device location tracking
Take traffic lights out of service
Manipulate sensor information
Bitcoin mining
Disable surveillance cameras

From a security perspective, a geographically dispersed system requires data communication, which typically has associated risks like confidentiality, integrity and availability. The diversity of technologies involved in an IoT system has the potential of introducing vulnerabilities as well.

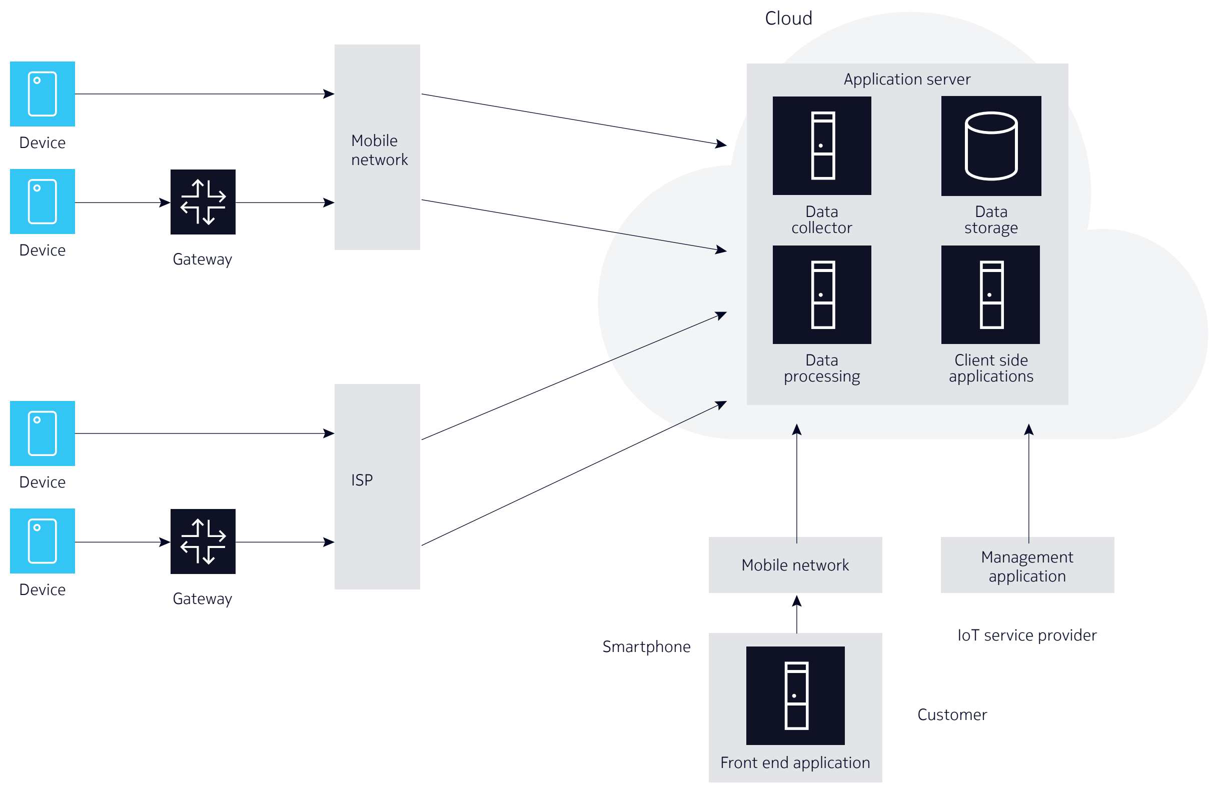
In order to secure an end-to-end IoT system, it is necessary to clearly understand the vulnerabilities and exploits associated with specific components or the system as a whole. This in turn requires an understanding of the architecture of an IoT system, the functionality provided by the components, the data and control flow across the system and the technologies and data communications involved.

This whitepaper provides a wealth of information that can help readers understand the current challenges of securing IoT systems and implementing security controls:

- a brief description of IoT systems;
- a comprehensive review of vulnerabilities associated with them;
- possible attacks against various components and the entire system;
- a review of the current trends in attacking IoT systems;
- predictions regarding the evolution of security threats;
- recommendations for securing IoT end-to-end solutions; and
- Nokia's approach to providing a comprehensive solution for IoT security.

This section presents the components of the IoT system and the functionality they provide.
The following diagram (figure 2) illustrates the high-level architecture of a typical IoT system.

Figure 2. High-level architecture of a typical IoT system



A device includes hardware and software that directly interact with the physical world. The leading role of these devices is to perform actions in the real world, acting on commands received from applications in the cloud. Devices are also able to collect telemetry, read-only data about the environment (usually collected through sensors), status data, etc. The devices sometimes provide data-processing functionality before sending the data to the cloud.

Devices usually communicate with each other through a local network. They also communicate to centralized applications in the cloud. Devices may be directly or indirectly connected to the Internet.

A gateway manages traffic between networks that use different protocols, providing for protocol translation and other interoperability tasks. The main role of an IoT gateway is to provide the connection and translation between devices and the cloud. When IoT devices don't support Internet connectivity (either because they don't have a direct Internet connection or they don't contain the network stack required for Internet connectivity), the gateway device acts as a proxy.

Other secondary tasks performed by IoT gateways are data aggregation from multiple devices for sending to the cloud over a single link, data store and forward, providing time synchronization and acting as a local cache for firmware updates.

The application servers in the cloud represent a central part of an IoT system. The cloud solution fulfills multiple roles:

- Collecting data from the IoT devices: The application server collects sensor data, telemetry data, operational information about the devices and the IoT infrastructure, events, etc. from the IoT devices in the field;

- Storing data: The cloud application server usually offers a range of storage solutions from unstructured blobs of data, to structured entity storage of devices or transactions, events and raw telemetry data;
- Processing data: Performing analytics on data obtained through IoT sources is an essential task, meant to generate higher value from the wealth of available raw data. Other tasks include data transformation, aggregation and enrichment; and
- Running client-side applications: These applications support the user in visualizing data and issuing commands to the IoT devices.

Front-end application represents another important component of an end-to-end IoT system. It provides the owner/operator of the IoT system with capabilities to manage the various IoT devices:

- Visualizing data (raw data, statistics, and higher-value data derived from the raw data) and events
- Issuing specific commands to the IoT devices

Management application implements operational management features. Operational management operations are typically provided by a management application that may be completely separate from the front-end application run by the end user. These applications may be run by the manufacturer or by the IoT service provider. The most important device management functions are:

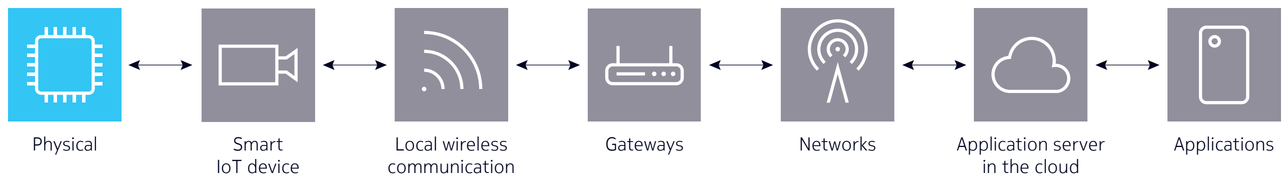
- performing inventory control
- performing maintenance operations (firmware upgrades, restarts, etc.)

IoT Vulnerabilities

As presented in the previous section, IoT systems tend to be complex and heterogeneous, including multiple layers, technologies, deployment locations, manufacturers, APIs, etc. From a security perspective, end-to-end IoT systems have numerous specific vulnerabilities. This section presents various aspects that need to be considered for a comprehensive approach to IoT security.

Smart IoT devices and IoT gateways

Figure 3.1. Vulnerabilities at the physical level

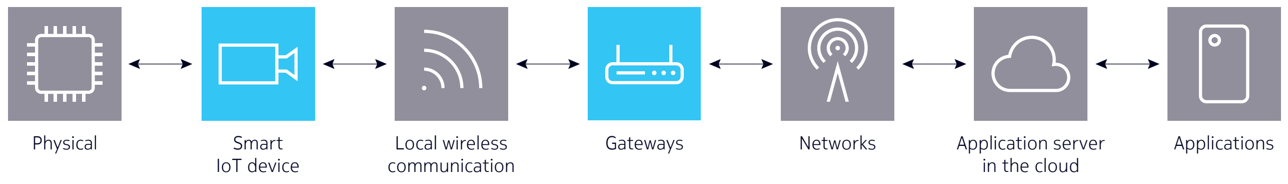


The problem of physically securing devices is specific to IoT, as some devices are located in public areas and components can be tampered with physically.

Orphaned devices and abandonware are easy to compromise and misuse. Once an IoT system is no longer in use, the operator should remove or disable the IoT devices, but this is not always the case. Since they are no longer actively managed, these devices may have vulnerabilities that were not dealt with.

Smart IoT devices and IoT gateways

Figure 3.2. Vulnerabilities at the smart IoT device and gateway level



The execution of applications on IoT devices is not initiated by user action, but is mostly done remotely. Therefore, hackers cannot convince the user to execute a piece of malware, as is the case with computers and handhelds, to take control of the computer. In order to take control of an IoT device, it is necessary to take advantage of vulnerabilities in sensor/device software or in application servers or gateways.

Because of the resource limitations, implementation of various stacks may be simplistic, offering less protection. Proper exception handling and proper input validation may be incomplete/deficient, opening the door for common exploits like buffer overflow. Especially the security of the communication channels can be compromised through the use of simplified protocol stacks.

Excessive direct exposure of IoT devices to the Internet represents a security risk. Regular computers are generally placed behind firewalls that block traffic initiated from the Internet. IoT devices (or gateways) need to be able to accept commands from the application server in the cloud and therefore need to respond to communication initiated by entities in the Internet. This increases the risk of being compromised by requests seeking to exploit the vulnerabilities of the devices.

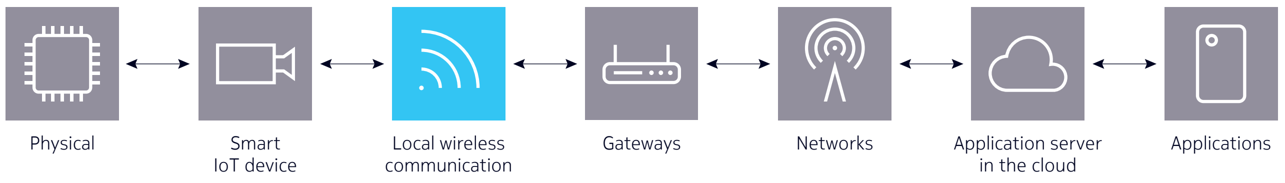
Post-installation support and firmware upgrades are often poor, either because they are impossible or inconvenient to carry out, or due to the fact that they are neglected or expired. Many of these devices are leaving the manufacturer in a weak state, and even if firmware updates are made available, few devices have an automated update mechanism. Configuration problems are also common, having no access or no capability to modify configuration or to adjust capabilities and functionality.

The application layer is most likely to be dependent on strong and efficient security features implemented in the transport layer. However, the protocol stack may not implement comprehensive support for authentication, authorization and encryption.

Software libraries reused during the development of firmware for IoT devices can introduce vulnerabilities in large numbers of IoT devices. Sometimes, these libraries have deficiencies from a software security perspective.

Local wireless communication

Figure 3.3. Vulnerabilities at the level of local wireless communication



Communication between IoT devices and gateways is normally done through non-IP protocols. These protocols are potentially less secure than the IP-based ones. Hacking these protocols is less likely, but probably easier than in the case of IP protocols, which are mature. One potential result is that the malware can spread between IoT devices directly through peer-to-peer communication.

Proximity-based security can be defeated using signal relays. In some IoT applications, proximity is used to authenticate the legitimate user. Even if the communication between devices is protected and cannot be tampered with, an attacker may gain an advantage by simply extending the range of the wireless communication. A typical example is the unlocking of a car through the simple presence of the key in the immediate proximity.

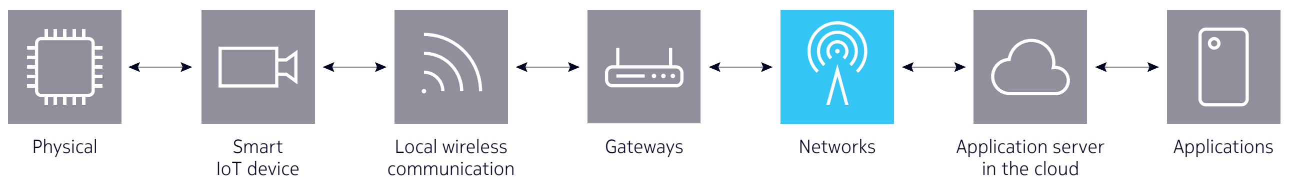
A local data link is prone to be implemented with proprietary protocols, lacking protection from common attacks (like man-in-the-middle or replay attacks), lacking encryption (favoring reverse-engineering or snooping).

Firmware upgrades can be hijacked, as this is sometimes done peer-to-peer. Especially for small IoT devices, manual configuration is expensive and over-the-air firmware and software updates are more convenient.

NOKIA

Communication between smart device/gateway and application server in the cloud

Figure 3.4. Vulnerabilities at the level of communication over Internet

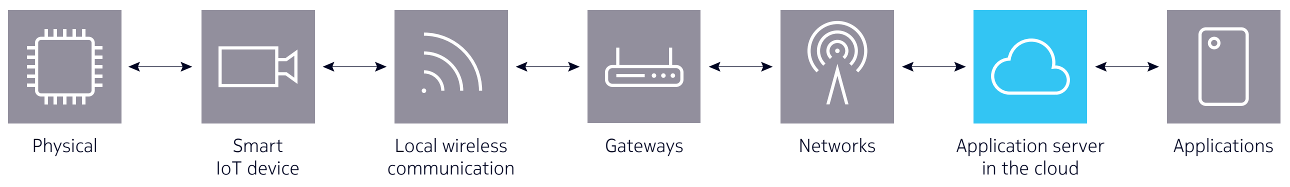


The problem of trust in data received from an element is of great importance in IoT systems. Either by having a rogue device sending fake data, or by altering the normal data communication flow, the data received by the consumer can be unreliable.

Sending unencrypted data over the Internet is a major concern, not only because data confidentiality is not provided, but also the packet sniffing can reveal credentials and detailed information about the communication protocol.

Application server in the cloud

Figure 3.5. Vulnerabilities at the application server level

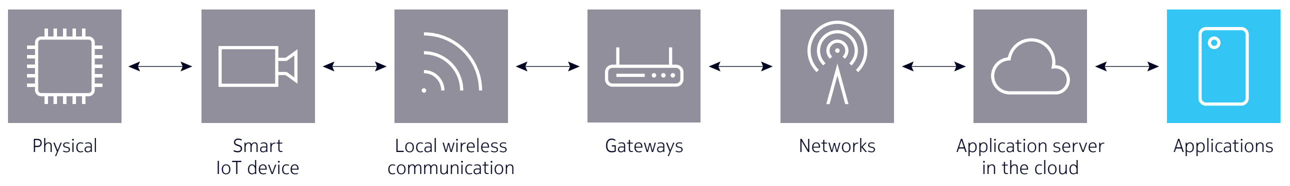


The application server represents a central point of vulnerability. By hacking it, unauthorized actors can have unlimited access to the data from multiple users.

Establishment of an IoT network is the process whereby entities need to join and be accepted as trusted elements of the network. There is the potential for masquerading, as rogue devices may be sneaking in as a supplier of invalid data or unauthorized consumer data.

Front-end applications (for end users and providers of IoT services)

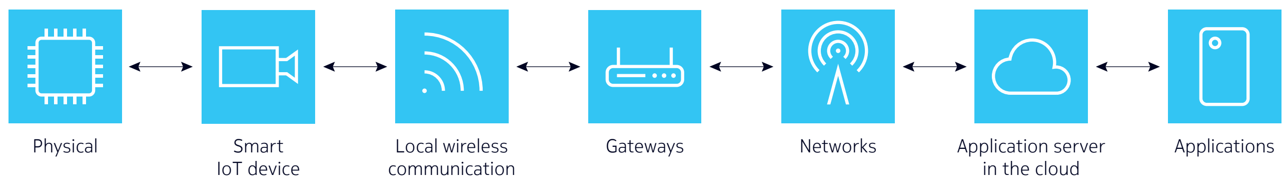
Figure 3.6. Vulnerabilities at the application level



It is necessary to protect handheld devices. There are countless ways to steal user credentials; once credentials are stolen, the entire end-to-end IoT network activity will appear legitimate, and any samples will be indistinguishable from the legitimate ones. A typical case is infection of the smartphone just to steal the credentials in order to access the front-end applications used by the customer to manage the IoT devices.

At the level of the entire IoT system

Figure 3.7. Vulnerabilities at the level of the entire IoT system



Because of the diversity of IoT entities and the need for integration, there is a trend to implement and publish an API that can be used to interact with the devices. However, public APIs offer hacking opportunities.

In order to manage the diversity of devices and technologies of IoT systems, as well as to allow easy deployment, middleware software is commonly used across an end-to-end IoT system. Hackers can target vulnerabilities of certain middleware software packages, compromising all entities using that technology.

Devices tend to be mass-produced, which makes it attractive to develop attacks to exploit the vulnerabilities of a specific device or even a specific product model. It is known that hackers invest in creating malware if the targeted devices are deployed in considerable numbers, which is the case with IoT devices.

Identification and authentication of devices is often inappropriate. It is not easy to set up or change keys/passwords, leading to heavy use of defaults that are easy to exploit.

Consumers tend to neglect the security of IoT devices they install and manage themselves. Consumers are less likely to secure their Internet-connected appliances than their own personal computer. This problem is aggravated by the fact that the owner of the device is not directly affected when the device gets infected. Typically, infected devices are used for mounting distributed denial of service (DDoS) attacks on targets in the Internet, creating problems not only for the target of the attack, but also for the ISP or wireless communication provider carrying the traffic.

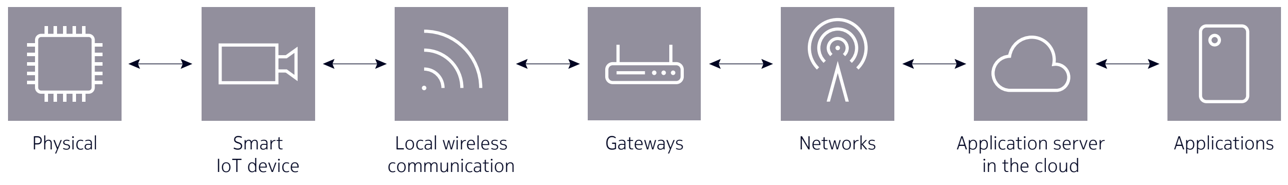
One hacked IoT device opens a backdoor to all devices connected to the local network (including the router), compromising a network assumed to be trusted. The IoT devices are particularly vulnerable to attacks from the Internet, due to the visibility and accessibility of the Internet and the remote command and configuration capability (acting as servers).

The emerging IoT market creates opportunity for malware creators and operators. With the proliferation of the IoT, the pool of possible botnet nodes is growing, enticing bad actors to invest resources in the development of IoT-specific malware. The numbers are in the attacker’s favor.

Overview of IoT exploits

An end-to-end IoT system encompasses multiple components, each of which can be subjected to specific attacks. This section summarizes the attacks that can affect each of the components.

Figure 4. Components of IoT systems that can be targets of attacks



Attacks on the physical device

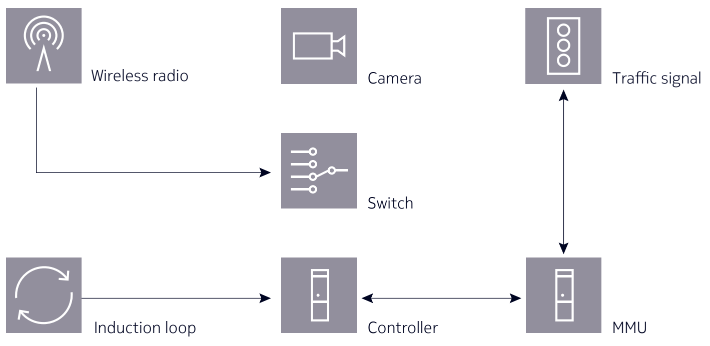
Some devices located in public spaces have poor physical security controls, making them vulnerable to tampering. Some devices are meant to work as part of an IoT device-management group. A common attack on these devices is to take over the device and assign it to another group. Other possible attacks on the physical devices are repurposing, destruction or invalidation.

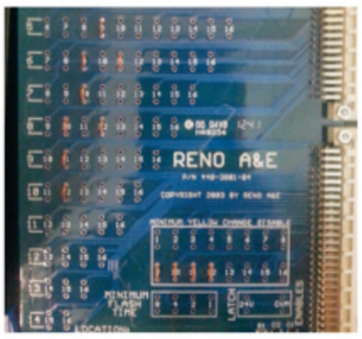
One example of physical tampering attacks is the targeting of the traffic infrastructure.

Currently deployed networked traffic signal systems pose a number of security flaws that can be leveraged to create attacks that gain control of the system. The most dangerous attacks are those that compromise the safety of the traffic signal systems remotely. Most security safeguards are geared toward thwarting such attacks. However, a physical attack on equipment located in public spaces can bypass all security mechanisms.

The modern traffic intersection is an amalgamation of various sensors, controllers and networking devices. Among them, there is a component that is dedicated exclusively to ensuring the safe operation of the traffic light system. The pictures below show a typical traffic intersection and the MMU configuration board.

Figure 5. Example of common traffic intersection (left) next to the MMU configuration board (right)





Malfunction management units (MMUs) are hardware-level safety mechanisms that act as a last line of defense against attacks. It ensures that the lights are not put in an unsafe configuration, putting the entire system into a known-safe configuration if any anomaly is detected. However, if the attacker gains physical access to the control system, it is easy to reconfigure the safe configuration setup, just by rewiring the valid configurations stored on the circuit board.

Attacks on the smart IoT devices

The most common attacks are related to identity, secure tokens and passwords. Other possible attacks on smart IoT devices are client-side certificates and key-related attacks. An example of an attack targeting a smart IoT device is an attack on web-based data acquisition systems primarily used for distributed PV solar metering and data acquisition. LGate is deployed across the energy sector and is used primarily in North America.

The Locus Energy meters use a PHP script to manage the energy meter parameters for voltage monitoring and network configuration. The vulnerability consists in the fact that the PHP code does not properly validate information that is sent in the POST request. If the web server port is publicly exposed, this vulnerability could be exploited remotely.

The command injection vulnerability allows hackers to hijack vulnerable solar meters.

A hacker could manipulate metering data to boost solar credits earned. An attacker could hack multiple systems to spoof power levels of solar arrays reporting back to a power grid. The hackers could enlist hijacked meters into DDoS botnets.

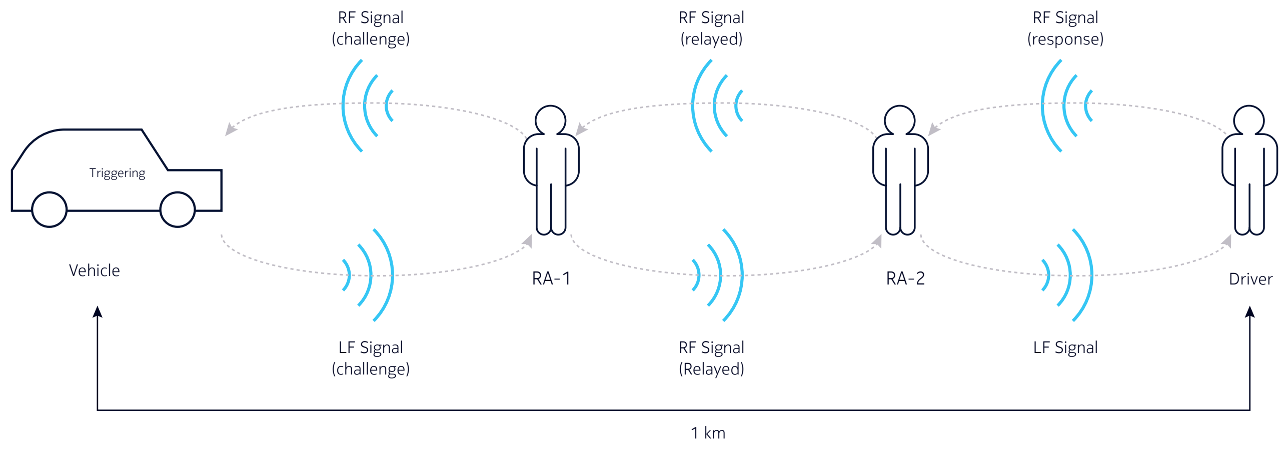
Local network attacks

There are a multitude of attacks targeting the communication between smart IoT devices and the gateway, including snooping, man-in-the-middle, take over, spoofing/simulation, redirection and command injection.

An example of an attack at this level affects multiple models of cars with keyless entry systems.

The vulnerability consists in the fact that the radio connection between keys and car can easily be extended over several hundred meters, regardless of whether the original key is. This vulnerability was exploited successfully. The diagram below illustrates a typical RF relay attack.

Figure 6. Example of typical radio frequency relay attack



The attack is performed by building a pair of radio devices; one is meant to be held a few feet from the victim's car, while the other is placed near the victim's key fob. The first radio impersonates the car's key and pings the car's wireless entry system, triggering a signal from the vehicle that seeks a radio response from the key. That signal is then relayed between the attackers' two radios as far as 300 feet, eliciting the correct response from the key, which is then transmitted back to the car to complete the "handshake".

In this way, the attacker can reliably unlock the target vehicles and immediately drive them away.

Gateway attacks

Gateways are exposed to attacks from the Internet, the most common of which are related to the gateway identity, secure tokens and passwords, and gateway certificates and keys.

There are multiple instances of attacks that target IoT gateway devices. Most of these attacks are credential compromise attacks, where the attacker attempts to guess the credentials and subsequently to log to the device while impersonating the legitimate owner. Mirai is a well-known malware that attempts to hijack as many devices as possible and to enlist them in a botnet.

Global network attacks

As the data traffic between IoT devices and the servers in the cloud passes thru the public Internet, it is subject to transport corruption, transport disruption and snooping attacks.

IoT data poisoning attacks can be performed after the attacker hijacks the devices and feeds fake data to the server. It is also possible that the attacker floods the application server with fake IoT data from other compromised devices.

Machine learning systems trained on user-provided data are especially susceptible to data poisoning attacks, as the false training data can corrupt the learned model.

Cloud service attacks

The application servers in the cloud can be subjected to various attacks, such as denial of service, snooping/probing, man-in-the-middle, take over and spoofing/simulation.

A vulnerability that was discovered in the SmartThinQ application (used to manage LG smart appliances) illustrates "take over" attacks, which can target a whole class of devices (in this case, all LG SmartThinQ smart home devices). The HomeHack vulnerability exposed millions of users of LG SmartThinQ smart home devices to the risk of unauthorized remote control of their SmartThinkQ home appliances.

The vulnerabilities in the SmartThinQ mobile app enabled Check Point's researchers to create a fake LG account, and then use this to take over a user's legitimate LG account, in turn gaining remote control of the user's smart LG appliances.

Once in control of a specific user's LG account, the attacker is able to control any LG device or appliance associated with that account, including robotic vacuum cleaners, refrigerators, ovens, dishwashers, washing machines, dryers and air conditioners.

Cloud application attacks

There are a multitude of attacks against IoT applications and the mobile devices on which they run, such data manipulation, resources access and abuse, take over, repurposing and identity.

It is an established fact that stolen passwords are the key to most data breaches (over 80% according to some estimates). Some types of malware specialize in stealing user login credentials of specific applications (social networking, banking, etc). While there are no known cases of stealing of credentials targeting specific IoT applications, there are types of malware that harvest this type of information (e.g. keyloggers). It is just a matter of time until IoT applications become the focus of cybercriminals. Accessing the applications that are used to manage IoT accounts represents one of the most severe security breaches, with serious implications.

Evolution of actual IoT exploits

The current trend is toward enlisting of large numbers of IoT devices and the creation of massive IoT botnets. The targeted devices are those with direct exposure to the Internet (not behind NAT firewalls).

The spread of malware doesn't usually involve a human factor. More than 80% of infections of smart devices and laptops occur through attacks where the device's user is tricked into running malicious applications. Since the IoT devices typically don't have a user interface, this method of malware dissemination is not applicable in the case of IoT devices.

The post-infection activity is also specific to IoT devices. Due to the limitations of the capabilities of IoT devices and the large number of devices that can be infected, the predominant use of IoT botnets is to execute DDoS attacks. The most common post-infection activities seen for non-IoT devices (stealing of credentials and sensitive information, advertisements, cybercurrency mining, etc.) are either not possible or not effective in the case of IoT devices.

The following sections present the evolution of IoT-specific malware in the direction of creating large botnets, which are subsequently used for DDoS attacks.

Knocking at the doors

The first step in the evolution of IoT botnets consisted of credential compromise attacks performed on the IT infrastructure (modems, routers), gateways or IoT devices. The goal is to penetrate the IoT system masquerading as a legitimate user. Once in the system, the actions of the hacker are indistinguishable from those of the legitimate user. There are multiple ways to perform credential compromise attacks:

- brute force attacks on device side hard-coded credentials, default credentials and deployment of default secure configurations
- dictionary attacks on devices, cloud services and applications using non-trivial passwords
- rainbow table attacks on devices, cloud services and applications relying on encrypted credential repositories

Mirai

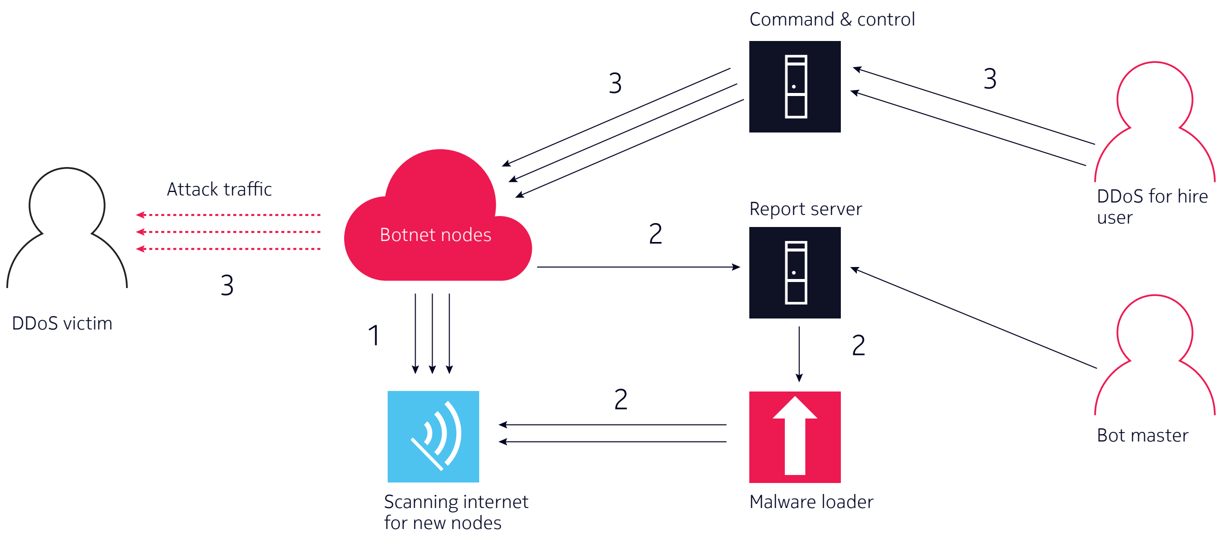
This malware first appeared at the end of 2016, but is worth mentioning, since it first highlighted the extent to which common IoT vulnerabilities (such as default/generic credentials) can be exploited to bring about devastating attacks. It also paved the way for ulterior attacks that were more powerful and sophisticated, which used basically the same principles.

In order to build the botnet, Mirai performs wide-ranging scans of IP addresses. The goal is to locate under-secured IoT devices that could be remotely accessed via easily guessable login credentials—usually factory default usernames and passwords.

Mirai's attack function consists mainly of launching HTTP floods and various network (OSI layer 3-4) DDoS attacks. For network layer assaults, Mirai is capable of launching GRE IP and GRE ETH floods, as well as SYN and ACK floods, STOMP (simple text-oriented message protocol) floods, DNS floods and UDP flood attacks.

The malware holds several killer scripts meant to eradicate other worms and Trojans, as well as prohibiting remote connection attempts of the hijacked device. The architecture and the workflows are worth mentioning, as they have been reused by other types of malware.

Figure 7. Diagram illustrates three distinct workflows used by malware: scanning; infection; and attack



Three distinct workflows are going on simultaneously:

- **Scanning** (step 1 in figure 6): The already infected devices perform SYN port scan to identify possible targets. A simple pattern matching is performed as part of the brute force authentication. The results of the scan are sent to a central reporting server.
- **Infection** (step 2 in figure 6): Based on the information from the report server, a specialized malware loader loads the executable that matches the architecture of the target. The newly infected device becomes immediately a member of the botnet and reports its infection to a command and control (C&C) server.
- **Attack** (step 3 in figure 6): The owner of the botnet issues a command to the C&C server, which instructs the botnet members to perform a specific attack action against the desired victim.

Hajime

Hajime is another IoT-based botnet, closely resembling the reconnaissance and infection behaviors of Mirai, but significantly more sophisticated. The main architectural difference is the fact that Hajime uses a peer-to-peer architecture instead of C&C server to send commands to bots. Hajime communicates over a distributed and decentralized overlay network to receive configuration and software updates.

The vigilante has no malicious functionality, the apparent goal being only to self-propagate and close down exposed telnet ports used by Mirai.

The remarkable characteristics of the Hajime botnet include:

- a variety of sophisticated cybertools;
- cross-platform compatibility capable of supporting five different platforms;
- a toolkit with automated tasks;
- a dynamic password list that could be remotely updated;
- option to download other codes, like brickerbot, even though this behavior has not yet been observed;
- detection evasion: monitors and learns traffic and behavior thresholds so that it can effectively mimic acceptable human behavior;
- embedded tool designed to remove firewall rules used to detect it;
- specific targeting of ISPs and MSSPs by identifying CPE devices and their CPE LAN Management Protocol; prohibits attempts to remove the rules that allowed the CPE device to talk to the service provider;
- use of multiple BitTorrent addresses that frequently change info hash or unique digital fingerprint;
- and extremely hard to defend against and next-to-impossible removal by external threats.

BrickerBot

BrickerBot's attack vector is similar to Mirai's, which turns infected devices into bots. The main difference is that BrickerBot "bricks" the device it infects, rendering the device permanently inoperable.

BrickerBot is a real-world instance of phlashing—or permanent denial of service (PDoS)—in which security flaws in the device's hardware are exploited and its firmware modified.

Like other malware, like Mirai and LuaBot (ELF_LUABOT), that target IoT devices, BrickerBot employs dictionary attacks (via default or hard-coded credentials) to gain unauthorized access to the device. However, BrickerBot is unlike other malware in the IoT threat landscape that typically amass the infected devices into botnets that can be leveraged in DDos attacks. BrickerBot instead executes a chain of malicious Linux commands that result in permanent damage in the device. Some of these commands include corrupting or misconfiguring the device's storage capability and kernel parameters, hindering Internet connection, tampering with device performance, and wiping all files on the device.

BrickerBot has two public versions:

- BrickerBot.1 targets IoT devices running BusyBox with exposed Telnet or Secure Shell (SSH), the latter due to having an older SSH server version. Networking devices running outdated firmware are the most at risk.
- BrickerBot.2 targets Linux-based devices with exposed Telnet service and default (or hard-coded) credentials. This second version employs TOR exit nodes to anonymize or conceal its activities. The malware also exploits remote code execution vulnerabilities in routers.

Breaking the doors

As the competition between the IoT botnets heats up (most notorious is the Mirai versus Hajime race to hijack IoT devices), new ways to hijack IoT devices were explored.

The first wave of IoT botnets took advantage of the improper configuration of IoT devices, particularly the failure to change the default password. Once a device was enlisted in a botnet, one of the first actions was to change the credentials, in order to keep other malware out.

Once the pool of IoT devices that are easily accessible from the Internet and have weak security credentials had been exhausted, the attention turned toward the exploitation of vulnerabilities that exist in the operating system, communication stacks, or at the level of applications. This threat trend was favored due to several factors:

- Simple computing devices: IoT devices typically have limited hardware and software capabilities when found in simple computing devices; vulnerabilities are common in software that has barely enough memory and processing resources to fulfill its basic functionality.
- Quick development: Some IoT devices are developed in a hastened period, which typically results in poor implementations, limited testing or quality assurance, and a lack of security controls.
- No upgrades: Many older versions of IoT devices were not designed to allow firmware upgrades or patching after initial deployment. Even if a device now has this capability, there are numerous factors that make the firmware upgrade impractical or impossible.
- Compromised devise abandonment: Since the infection of IoT devices rarely directly affects the owners of IoT devices, they have little incentive to go through a complicated process of restoring their IoT devices. There is often no urgency to restore IoT devices after infections.
- Mass infection: IoT devices are mass-produced which often means that any vulnerability in a particular model can be easily replicated and deployed, or even spread through various channels, creating a widespread and unmanageable infection.

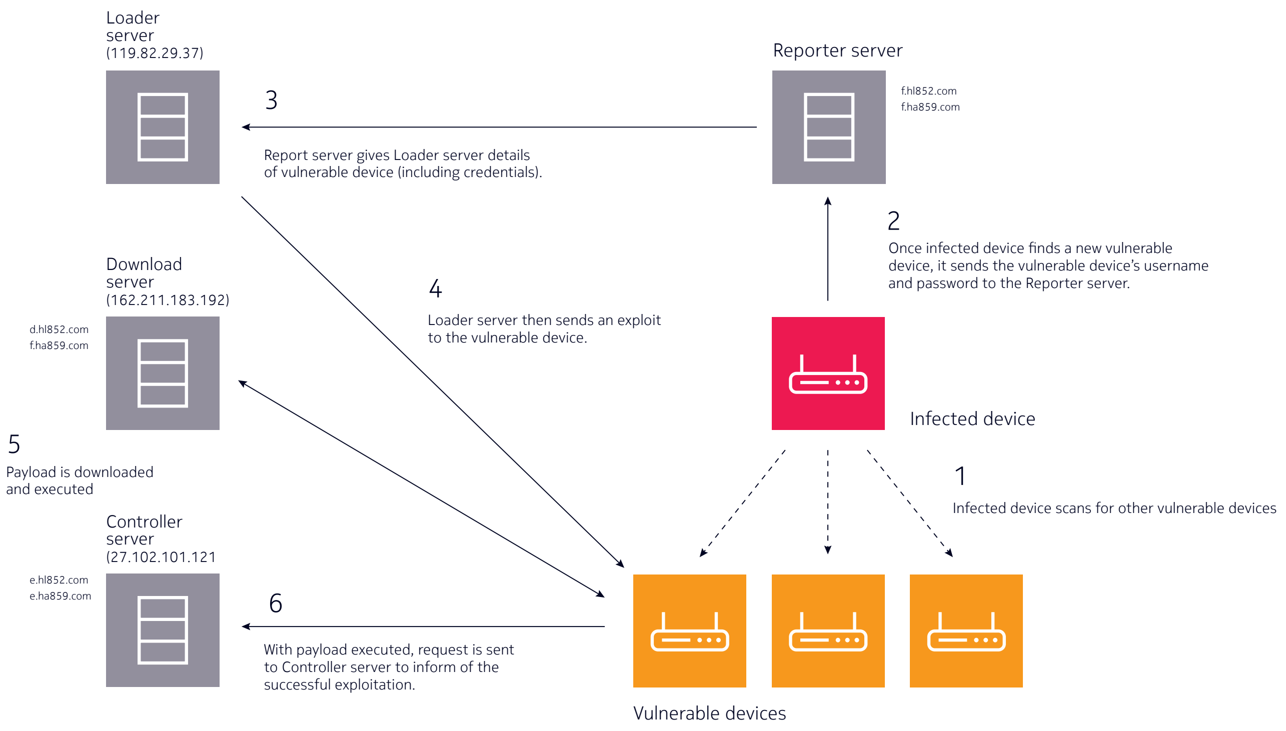
As a consequence, it is fairly easy to find vulnerabilities that can be exploited in a large number of devices. In this case, once a vulnerability is exploited and the IoT device is enlisted in a botnet, the offending malware cannot close the vulnerability, so the vulnerability can be exploited multiple times by different versions of malware.

In the following section we present overviews of three different botnets that spread by exploiting various vulnerabilities of IoT devices.

Reaper

A year after the appearance of the Mirai botnet, an even more sophisticated IoT-based attack took place. A vulnerability scanning functionality has replaced the original brute forcing password functionality as presented by Mirai. This functionality has been added to enable the infection of a greater number of devices using fewer resources, while reducing the malware detection rates.

Figure 8: The mechanism for the malware propagation is illustrated in the diagram below (source: Check Point Research)


Loader server (119.82.29.37)
Reporter server
f.hl852.com
f.ha859.com
3
Report server gives Loader server details of vulnerable device (including credentials).
2
Once infected device finds a new vulnerable device, it sends the vulnerable device's username and password to the Reporter server.
Download server (162.211.183.192)
d.hl852.com
f.ha859.com
4
Loader server then sends an exploit to the vulnerable device.
Infected device
5
Payload is downloaded and executed
1
Infected device scans for other vulnerable devices
Controller server (27.102.101.121
e.hl852.com
e.ha859.com
6
With payload executed, request is sent to Controller server to inform of the successful exploitation.
Vulnerable devices

The mechanism used to infect IoT systems is based on the mechanism observed in Mirai, with a number of differences:

- The scan behavior is not very aggressive, so it can spread without drawing attention.
- The C&C server has been completely redesigned to operate with a new backend.
- The C&C communication protocol was changed as well and is unique to this malware.

The main improvement over both Mirai and Hajime is the fact that it is armed with exploits covering nine different known specific firmware vulnerabilities spanning a variety of IoT vendors (NetGear, Linksys, GoAhead and Avtech). The malware conducts a set of vulnerability tests on each of the generated IP addresses. These tests locate vulnerabilities in any of the following devices and/or infrastructure:

- Dlink https://blogs.securiteam.com/index.php/archives/3364
- Goahead https://pierrekim.github.io/blog/2017-03-08-camera-goahead-0day.html
- JAWS https://www.pentestpartners.com/blog/pwning-cctv-cameras/
- Netgea https://blogs.securiteam.com/index.php/archives/3409
- Vacron NVR https://blogs.securiteam.com/index.php/archives/3445
- Netgear http://seclists.org/bugtraq/2013/Jun/8
- Linksys http://www.s3cur1ty.de/m1adv2013-004
- Dlink http://www.s3cur1ty.de/m1adv2013-003
- AVTECH https://github.com/Trietptm-on-Security/AVTECH

Once a device is infected, it constantly polls for available commands from the controlling C&C server. Once a command is received, a plain download, or a download followed by execution is performed.

The malware is built around a Lua engine combined with scripts used to run its attacks. Its attack code can be easily updated to include more malicious options.

The malware has several weaknesses, indicating that it is still in its early stages of development:

- It doesn't protect infected devices from being infected by other pieces of competing malware.
- Its control servers rely on static domain names and IP addresses, and it communicates over unencrypted HTTP channels.

This malware has the potential of posing a serious threat. The botnet's most innovative attribute is its exploit mechanism, which targets specific firmware vulnerabilities in a host of widely used devices. As the list of vulnerabilities exploited by this malware grows, the reach can increase significantly. The only effective countermeasure that could limit the expansion of such a malware would be firmware upgrades/patches to close the vulnerabilities.

Satori

In early December 2017, a new botnet named Satori affected 280,000 IP addresses in just 12 hours, infecting numerous home routers to become part of its botnet.

Instead of using a scanner to search for routers with default/generic credentials, the botnet uses two exploits that attempt to connect with devices on ports 37215 and 52869. In this way, it was able to infect even routers secured with strong passwords.

The malware can propagate rapidly by itself, which essentially makes it an IoT worm.

The new malware exploits two vulnerabilities in routers that are in widespread use:

CVE-2017–17215 — a vulnerability in Huawei Home Gateway routers (Huawei HG532), targets port 37215.

CVE-2014-8361 — a command injection vulnerability in Realtek SDK miniigd Universal Plug and Play (UPnP) SOAP, targets port 52869.

As in the case of the Mirai botnet, the source code was released.

Deutsche Telekom

An attack on customers of Deutsche Telekom in Germany and Eircom in Ireland highlighted again the major impact of exploiting a single vulnerability of a device that is in widespread use.

The massive Deutsche Telekom attack of last November that affected 900,000 DSL router customers' Internet, phone and video services was enabled by the TR-069 vulnerability. Attackers can exploit the issue and cause a device to download and execute an attack that, in this case, prevented the routers from resolving domains.

The malware exploited a then largely unknown flaw in routers that ISPs provided to their customers, taking advantage of a flawed implementation of router maintenance features implemented by two Taiwanese router manufacturers: Arcadyan Technology and Zyxel.

It took more than 24 hours to identify the cause of the router crashes and to patch most of the vulnerable routers.

Predictions

Lateral movements to compromise assets within the security perimeter

As IoT devices tend to be connected to private LANs with other sensitive devices, a large enough mass of compromised IoT devices can see other uses, such as mass-surveillance by eavesdropping on LAN traffic, exfiltration of confidential data, and monetary theft by capturing financial information.

One of the characteristics of the IoT devices is that they need to be accessible from the Internet. This is due to the need to send commands to the devices from servers in the cloud. This brings about an increased exposure to attacks coming from the Internet.

Currently, infected IoT devices can spread to other IoT devices. But what happens if infected IoT devices are able to infect non-IoT devices? This theory should be tested and validated to learn the signs of threats and enable a line of defense.

This would increase the number of devices that are part of the botnet, for sure. However, more worrisome is the fact that the IoT devices become the entry points through the cybersecurity perimeter. The danger is magnified by the impossibility of deploying significant endpoint security controls at the level of the IoT devices (e.g. antiviruses), which could stop the propagation of the malware.

Regarding the attacks perpetuated by IoT botnets, they are currently limiting themselves to unleashing DDoS attacks to selected targets. But what if the botnets turn their attention toward other activities, like spying? Especially if the IoT malware moves laterally to more significant targets within the compromised networks, the IoT botnets can morph into ideal spying tools.

Another possible move for IoT malware is to attack the IT infrastructure of the compromised network. The perimeter routers and firewalls are hardened to resist attacks from the Internet side, but are usually vulnerable to attacks from within the network.

Social compromise attacks

Social compromise attacks can be executed against the mobile devices that the end users use to manage their IoT account (for data visualization, monitoring, configuring, controlling). Handhelds used to control IoT devices represent the point in the IoT ecosystem where the human factor vulnerabilities can be most effectively exploited.

There are numerous ways in which a malware can take control over a handheld, such as running a Trojan application from a third-party app store or being the victim of a phishing attack, to name only a few.

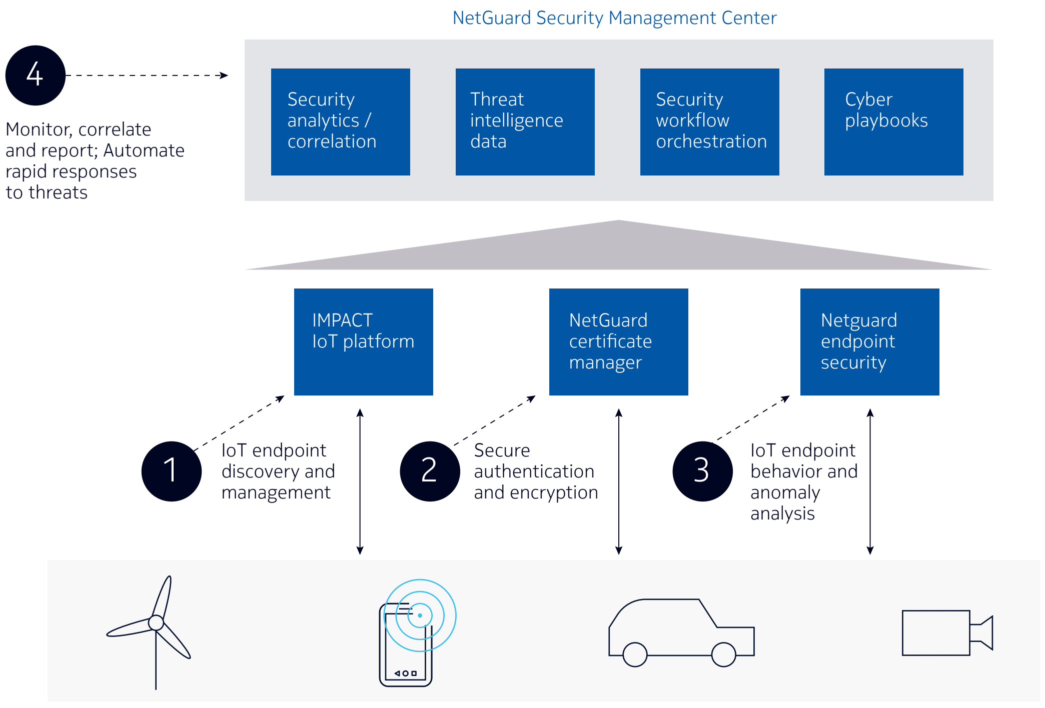
Increased use of AI in cybersecurity

Since 2017, the use of AI in cybersecurity has gained a lot of traction, and the expectation is that the usage of artificial intelligence and machine learning will further expand. So far, AI technologies have been employed in detection and protection mechanisms. However, it is predicted that cyber criminals will also use AI and ML to conduct attacks. For cyber criminals, the exploration of victim's networks and choosing the most appropriate ways to attack those networks represent the most labor-intensive part of their activity. This is especially pertaining the IoT domain, where the IoT components are so diverse. AI can help with the spread of malware on a wide variety of IoT devices, dramatically increasing the size of the botnets.

Nokia's approach to IoT Security

As the IoT systems encompass a multitude of components, such as IoT devices, local networks, gateways, global network (Internet), cloud services and cloud applications, it is difficult to implement a comprehensive security solution that can effectively deal with attacks against all the components, however Nokia's NetGuard Security Management Center solution, shown in the figure 9, does just that.

Figure 9. Diagram of Nokia's NetGuard Security Management Center


NetGuard Security Management Center
4
Monitor, correlate and report; Automate rapid responses to threats
Security analytics / correlation
Threat intelligence data
Security workflow orchestration
Cyber playbooks
IMPACT IoT platform
NetGuard certificate manager
Netguard endpoint security
1
IoT endpoint discovery and management
2
Secure authentication and encryption
3
IoT endpoint behavior and anomaly analysis

1. Nokia's IMPACT IoT platform provides secure device management. It allows the operator to deploy and manage IoT devices in a secure fashion, providing a secure configuration and providing security patches and firmware updates as required.
2. NetGuard Certificate Management combines with the IMPACT IoT platform to provide strong authentication and secure communication.
3. NetGuard Endpoint Security provides 24/7 monitoring to ensure that the devices are performing normally and have not been compromised by malware or attacks. This agentless approach to monitoring for malware activity works for both managed and unmanaged IOT devices.
4. NetGuard Security Management Center provides a single control point for security analysis and automated threat response for the entire IOT network.

NetGuard Endpoint Security (NES)

NetGuard Endpoint Security (NES) is a key component in Nokia's end-to-end security portfolio for IoT systems. It is deployed in the carrier network and provides 24/7 monitoring of the network traffic between the endpoint devices and the Internet. The system uses deep packet inspection and intrusion detection technologies to detect evidence of malware infection in this network traffic. This includes:

- botnet C&C traffic
- exploit attempts
- network scanning
- hacking activity
- DDoS activity
- data exfiltration
- anomalous behavior

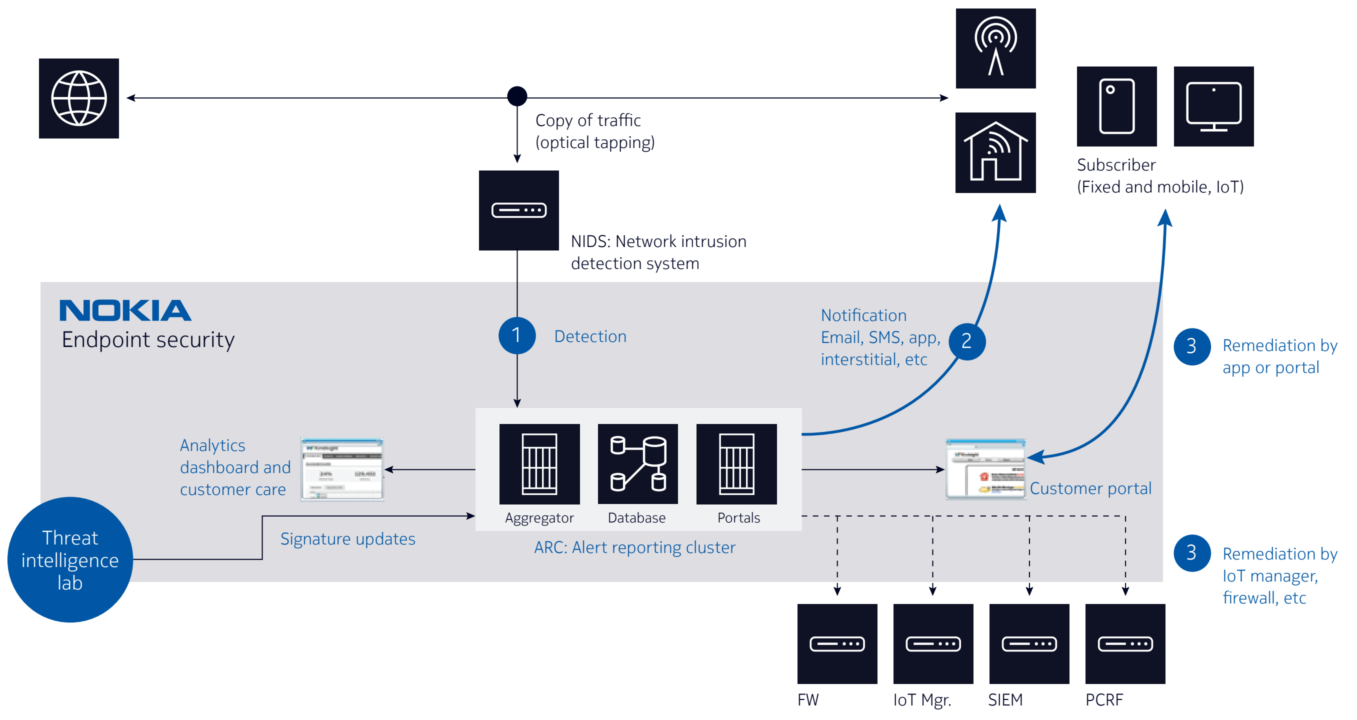
The system also uses machine learning to characterize the traffic patterns that are expected from specific IOT device types and will raise an alarm whenever an IOT device steps outside this normal traffic profile. This provides the ability to detect zero-day compromise of IOT devices.

The NES network-based security product portfolio has a proven record of enabling service providers and mobile network operators to detect and analyze threats, send alerts, block infected devices and protect subscribers. The NES system is designed for security analysts, system administrators, and help-desk personnel. The NES features are displayed in web-based GUIs, called portals, which can be used by:

- security and network analysts to monitor the ongoing operation of the system
- help-desk representatives to assist subscribers who use the system
- subscribers to monitor the status of their mobile devices and household network

NES Architecture

Figure 10: The following figure presents the high-level architecture of the NES system


Copy of traffic (optical tapping)
NIDS: Network intrusion detection system
Subscriber (Fixed and mobile, IoT)
NOKIA Endpoint security
1 Detection
Notification Email, SMS, app, interstitial, etc
2
3 Remediation by app or portal
Analytics dashboard and customer care
Aggregator
Database
Portals
Customer portal
Threat intelligence lab
Signature updates
ARC: Alert reporting cluster
3 Remediation by IoT manager, firewall, etc
FW
IoT Mgr.
SIEM
PCRF

The main system components of NES are the following:

- NIDS (Sensor server): Sensors that passively monitor traffic using a tap or mirror port on a router in the network
- ARC: System components, including the data aggregators, database storage, data-processing engines, and web servers that host web-based portals
- Web portals: Provide graphical user interfaces that give access to NES features
  - Security analytics: Service provider network security specialists monitor malware infections, provide subscriber alerts and generate reports.
  - Help desk: Subscriber support and help-desk personnel monitor subscriber alerts and assist subscribers in resolving malware infections.
  - Subscriber security: Subscribers access information about their infection status. The subscriber portal includes a self-service remediation feature that guides subscribers through the process of resolving infections.

The system also provides interfaces to policy enforcement devices such as firewalls and PCRFs that allow infected devices to be blocked or quarantined. In addition to Nokia's IMPACT IoT management system, it can also communicate infection information to third-party IoT system. NES also exports the infection information to Nokia's SMC and third-party SIEM.

IoT Specifics

NES provides excellent capabilities for the detection of malware activity at the level of IoT devices, gateways and communication infrastructure (e.g. modems and routers).

Detection of existing infections of IoT devices and IoT gateways is supported mainly through detection of communications with the C&C servers. One specific feature of IoT malware is that the attacks are launched not only from Internet sites, but also from individual IoT devices that are already infected. NES has the capability of detecting attacks perpetuated by infected IoT devices and notifying the subscriber associated with the infected device. This capability is very valuable, as the infected device most likely will not display obvious signs of infection and the device owners may not be aware of the attacks launched by their devices.

NES has strong capabilities in detecting attacks on the IT infrastructure (modems, routers, etc.). Most of the Internet-driven attacks target routers, such as a D-Link 850L router or Linksys E1500 Wireless-N router, as demonstrated by recent IoT botnets like Reaper. NES provides signature-based detection of incoming attacks from the Internet or from other IoT devices.

NES has also the capability of detecting DDoS attacks in which the IoT devices connected to the mobile network are participating.

An advanced feature of NES is behavior-based detection of malware activity. NES is able to learn the communication patterns of IoT devices and automatically build communication profiles. Later, NES applies this knowledge to detect deviations from normal behavior. In the simplest case, NES will raise alerts if the device suddenly starts communicating with unusual destinations.

NES provides extensive support for the detection of malware activity at the level of mobile devices. Handheld devices are commonly used for the remote management of IoT accounts, including visualization of data and issuing of commands. However, handheld devices are subject to attacks that target the user of the device (e.g. phishing attacks, downloading of Trojan applications from third-party app stores, etc.). As a result of the infection, the credentials used to access IoT accounts can be stolen, enabling the attacker to access the account.

NES has up-to-date intelligence about various types of malware that are known to steal account credentials and is able to detect infection with malware in a timely manner. If the end user is notified of the infection, the user can take simple but effective measures such as changing account passwords.

Recommendations

Securing IoT systems presents unique characteristics and challenges. The propagation of malware depends much less on the vulnerabilities associated with the human factor and much more with the hardening of all components of an end-to-end IoT system.

The operator/end user of IoT systems bears a lot of responsibility for the proper setup and configuration of the IoT system. Nokia's IMPACT line of products offers excellent support for the management, configuration and communication of all components of an IoT system. It represents an essential tool for providers of enterprise IoT services, covering all aspects of setup and management and maintenance of large IoT deployments.

The post-infection activity will likely not directly affect the owner of the infected IoT devices, as the IoT devices are usually used to launch DDoS attacks against distant targets. For enterprise IoT service providers, however, the effects of post-infection activity are bound to be significant, as the quality of the provided service is affected (degraded service, service interruptions, etc.).

Even if the infection of IoT devices doesn't usually directly affect the owners of the infected devices, it is still essential to detect malware propagation and post-infection activity. Nokia's NES network-based security product portfolio provides support to the operators and end users of IoT systems with the detection of attacks and malware propagation activities.

Best practices

The following recommendations highlight areas in which the operators/end users of IoT systems can take action to prevent successful attacks on their system. In most cases, Nokia products are available to support the operators/end users in their actions.

- Physically protect the IoT infrastructure: The worst security attacks against an IoT infrastructure are launched by gaining physical access to devices.
- Inventory control: Operators of an IoT system need to keep track of the kinds of IoT devices on their network. This includes the initial configuration of the IoT system and the subsequent monitoring (to make sure that no unauthorized devices join the system). Nokia's IMPACT system is specialized in device management, providing strong support for configuring IoT systems and monitoring deployed devices.
- Keep systems up to date: IoT system operators are responsible for ensuring that device operating systems and all device drivers are updated to the latest versions. Much of the overall vulnerability of IoT systems is due to the fact that deployed devices are not actively and timely upgraded and/or patched. Nokia's IMPACT system offers support for the firmware upgrade of the managed IoT devices to the latest version. IMPACT automates the process of checking for updates and applying patches when they become available.
- Protect cloud credentials: Cloud authentication credentials used for configuring and operating an IoT deployment are the first avenue for malware to gain access and compromise an IoT system. Nokia's IMPACT manages the configuration of login credentials and provides the framework for secure communication to and from IoT devices. NES specializes in detecting infections of mobile devices that can lead to the stealing of IoT credentials.
- Monitor outbound and lateral IoT traffic: It is important to monitor IoT devices for aberrant behavior in order to automatically identify rogue devices. Nokia's NES product offers the capability to baseline the normal traffic of IoT devices, then monitors for unusual behavior and notifies the user/operator

when such behavior is detected. Nokia's IMPACT product works together with NES and takes actions to remediate the situation (e.g. by restoring the firmware of the affected devices).

- Protect against malicious activity: The use of the latest anti-virus and anti-malware capabilities on each device is unfortunately not realistic for IoT devices. Due to the limited resources of IoT devices, it is not possible to implement elaborate security solutions at the device level. Nokia's NES and IMPACT products approach the security of IoT devices from a different perspective:
  - The devices are hardened by proper configuration and firmware upgrade to the latest version. IMPACT offers extensive capabilities in this direction.
  - The infection of IoT devices is detected in real time. NES detects attacks and infections on IoT devices as well as mobile devices used to manage the IoT system. NES automatically creates profiles based on the normal traffic of the managed IoT devices and detects deviations from those profiles.
  - Infected IoT devices are cleaned. IMPACT acts on the intelligence received from NES and takes action to restore the infected device, by rebooting or restoring the firmware.
- Segment the IoT traffic: It is important to keep the IoT traffic separate from other network segments to limit exposure and the spread of malware. It is proven that network segmentation and the usage of NAT play an essential role in impeding the initial attacks on a network and the spread of malware within a network.
- Audit frequently: Auditing IoT infrastructure for security-related issues is key when responding to security incidents. Both NES and IMPACT provide support for auditing the activity in the IoT system.

Standardization efforts

The IoT market is set to grow dramatically and security concerns are forcing governments to regulate IoT device manufacturers, requiring them to add stronger security to their products.

In Europe, the European Union Agency for Network and Information Security (ENISA) identifies and analyzes existing IoT security practices, security guidelines, relevant industry standards and research initiatives in the area of IoT security for critical information infrastructures (such as Industry 4.0, M2M communications, IoT updatability).

By comparing existing practices and standards, ENISA developed baseline security measures to be adopted by relevant stakeholders. The main focus is on IoT resilience and communication, the interoperability with proprietary systems and the reliability of IoT.

The set of security measures/good practices in this report has been determined based on extensive and thorough desktop research, which took into account different security guidelines, standards, etc.

ENISA identifies different security measures and good practices for the following security domains:

- Information system security governance and risk management: Includes security measures regarding information system security risk analysis, policy, accreditation, indicators and audit, and human resource security.
- Ecosystem management: Includes security measures regarding ecosystem mapping and ecosystem relations.
- IT security architecture: Includes security measures regarding system configuration, asset management, system segregation, traffic filtering and cryptography.

- IT security administration: Includes security measures regarding administration accounts and administration information systems.
- Identity and access management: Includes security measures regarding authentication, identification and access rights.
- IT security maintenance: Includes security measures regarding IT security maintenance procedures and remote access.
- Physical and environmental security.
- Detection: Includes security measures regarding detection, logging, and log correlation and analysis.
- Computer security incident management: Includes security measures regarding information system security incident analysis and response, and incident reports.
- Continuity of operations: Includes security measures regarding business continuity management and disaster recovery management.
- Crisis management: Includes security measures regarding crisis management organization and process.

The identified IoT baseline security measures are presented according to three main categories:

- Policies (PS), covering security by design, privacy by design, asset management and risk and threat identification and assessment.
- Organizational, people and process measures (OP), covering end-of-life support, proven solutions, management of security vulnerabilities and/or incidents, human resources security training and awareness and third-party relationships.
- Technical measures (TM), covering hardware security, trust and integrity management, strong default security and privacy, data protection and compliance, system safety and reliability, secure software/firmware updates, authentication, authorization, access control: physical and environmental security, cryptography, secure and trusted communications, secure interfaces and network services, secure input and output handling, logging, monitoring and auditing.

In the United States, The Internet of Things Cybersecurity Improvement Act of 2017, was introduced on August 1, 2017 by four U.S. senators; it was developed in response to a series of IoT-related cyber-attacks that took place in 2016. This act establishes minimum cybersecurity requirements for connected devices purchased by the U.S. government.

Among the imposed requirements, the most significant ones are the following:

- Requiring vendors to ensure that their devices are patchable, rely on industry-standard protocols, do not use hard-coded passwords, and do not contain any known security vulnerabilities;
- Requiring vendors selling IoT devices "to provide written certification that the device does not contain, at the time of submitting the proposal, any hardware, software, or firmware component with any known security vulnerabilities or defects." If a vendor identifies vulnerabilities, it must disclose them and patch them in a timely manner;
- Requiring each executive agency to inventory all Internet-connected devices in use by the agency;
- Along with National Institute of Standards and Technology (NIST), specifying particular measures, e.g. network segmentation, for agencies to employ them;

- Directing the Department of Homeland Security's (DHS) National Protection and Programs Directorate (NPPD) to develop coordinated disclosure guidelines, allowing researchers to uncover vulnerabilities in and share them with the vendors; and
- Requiring an effectiveness report, with recommendations for updates, to be submitted to Congress after five years.

The NIST IoT Security Guidance document (special publication 800-160) is designed to help prevent the vulnerabilities that lead to their exploitation and to facilitate "a disciplined, structured, and standards-based set of systems security engineering activities." To accomplish this, the Guidance focuses on assessing the trustworthiness of various Internet-connected devices and their impacts through a series of processes governed by the life cycle of each device, breaking those processes into four categories:

1. Agreement processes
2. Organization-project enabling processes
3. Technical management processes
4. Technical processes

In each of these categories, special publication 800-160 uses international system engineering standards and maps out the purpose, outcomes and various activities and tasks associated with each life-cycle process. It addresses the activities and tasks, the concepts and principles, and most importantly, what needs to be considered from a security perspective when executing within the context of systems engineering.

Together with other NIST special publications addressing security, the Guidance emphasizes the need to proactively consider and develop security as a part of the product design process and to consider it throughout both the design process and the product's life cycle.

About the Nokia Threat Intelligence Center

The Nokia Threat Intelligence Center focuses on the behavior of malware network communications to develop detection rules that identify malware infections based on command-and-control communication and other network behavior. This approach enables the detection of malware in the service provider's network and the detection rules developed form the foundation of the Nokia network-based malware detection product suite.

To accurately detect that a user is infected, our detection rule set looks for network behavior that provides unequivocal evidence of infection coming from the user's device. This behavior includes:

- Malware command-and-control (C&C) communications
- Backdoor connections
- Attempts to infect others (for example, exploits)
- Excessive email
- Denial of Service (DoS) and hacking activity.

Four main activities support our signature development and verification process:

- Monitoring of information sources from major security vendors and maintaining a database of current and active threats

- Collecting malware samples (>200,000/day), and classifying and correlating them against the threat database
- Executing samples that match the top threats in a sandbox environment and comparing them against our current signature set
- Conducting a detailed analysis of the malware's behavior and building a new signature, if a sample fails to trigger a signature.

Find out more about the Nokia Threat Intelligence Center, visit our Security solution page, or learn more about the Nokia NetGuard Endpoint Security solution.

Abbreviations

API	Application programming interface
Botnet	A group of computers connected in a coordinated fashion for malicious purposes
C&C	Command and control
DDoS	Distributed denial of service
ENISA	European Union Agency for Network and Information Security
GRE	Generic routing encapsulation
IoT	Internet of things
ISP	Internet service provider
M2M	Machine-to-machine
MMU	Malfunction management units
MSSP	Managed security service provider
NAT	Network address translation
NES	NetGuard Endpoint Security
NIST	National Institute of Standards and Technology
PCRF	Policy charging and rules function
PHP	Hypertext preprocessor is a server-side scripting language designed for web development but also used as a general-purpose programming language
PDoS	Permanent denial of service
POST	Submits data to be processed to a specified resource
PV	Photovoltaic
SSH	Software package that enables secure system administration and file transfers over insecure networks
SIEM	Security information and event management system
STOMP	Simple text-oriented message protocol
SYN-ACK	The method used by TCP set up a TCP/IP connection over an Internet Protocol-based network
UDP	User datagram protocol
TCP	Transmission control protocol
TOR	The onion protocol, directs Internet traffic through a free, worldwide, volunteer overlay network consisting of more than 7,000 relays to conceal a user's location and usage