



Home (<https://embedi.com>) / Blog (<https://embedi.com/blog>) /  
Analytics (<https://embedi.com/blog/categories/analytics/>) /  
Grim IoT Reaper: 1- and 0-day vulnerabilities at the service of botnets

## Categories:

- Analytics (<https://embedi.com/blog/categories/analytics/>)
- Research (<https://embedi.com/blog/categories/research/>)

## Tags:

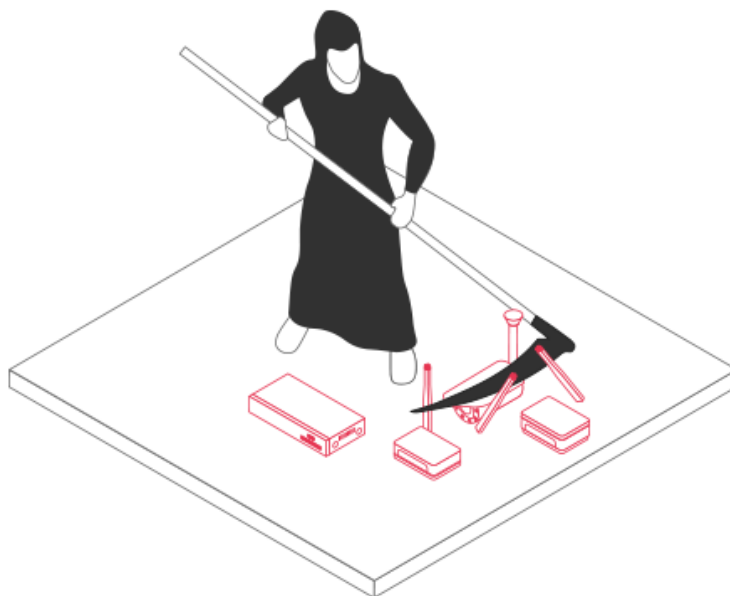
#ATM (<https://embedi.com/blog/tags/atm/>)    #CISCO (<https://embedi.com/blog/tags/cisco/>)  
#cybersecurity (<https://embedi.com/blog/tags/cybersecurity/>)  
#D-Link (<https://embedi.com/blog/tags/d-link/>)    #DJI (<https://embedi.com/blog/tags/dji/>)  
#exploitation (<https://embedi.com/blog/tags/exploitation/>)  
#firmware-security (<https://embedi.com/blog/tags/firmware-security/>)  
#hardware (<https://embedi.com/blog/tags/hardware/>)  
#hijacking (<https://embedi.com/blog/tags/hijacking/>)    #intel (<https://embedi.com/blog/tags/intel/>)  
#Microsoft (<https://embedi.com/blog/tags/microsoft/>)    #mobile (<https://embedi.com/blog/tags/mobile/>)  
#Office (<https://embedi.com/blog/tags/office/>)    #RCE (<https://embedi.com/blog/tags/rce/>)  
#router (<https://embedi.com/blog/tags/router/>)    #SCADA (<https://embedi.com/blog/tags/scada/>)  
#vulnerabilities (<https://embedi.com/blog/tags/vulnerabilities/>)

## Popular articles:

Killchain of IoT Devices. Part 2 (<https://embedi.com/blog/killchain-iot-devices-part-2/>)  
[Killchain of IoT Devices. Part 1 \(https://embedi.com/blog/killchain-iot-devices-part-1/\)](https://embedi.com/blog/killchain-iot-devices-part-1/)

30 January, 2018

# Grim IoT Reaper: 1- and 0-day vulnerabilities at the service of botnets



Category: Analytics (<https://embedi.com/blog/categories/analytics/>)

On the 19th of October, 2017, the world of IoT shuddered, facing a new enemy – a huge botnet that would be later called Reaper.

Reaper is grim and is by far grimmer than the notorious Mirai botnet. According to the data provided by 360 Netlab ([http://blog.netlab.360.com/iot\\_reaper-a-rappid-spreading-new-iot-botnet-en/](http://blog.netlab.360.com/iot_reaper-a-rappid-spreading-new-iot-botnet-en/)), it has already infected approximately 30,000 smart-devices, and about 2,000,000 are still vulnerable.

No wonder. Mirai exploited weak and default passwords. Reaper's scythe is more sharp-edged because the botnet exploits dozens of vulnerabilities in different IoT devices, including routers and cameras produced by D-Link, Wi-Fi CAM, JAWS, Netgear, Linksys, AVTECH, and others. And, with time it grows even sharper: the botnet is expanding while owners of IoT devices keep their gadgets outdated.

The vulnerabilities Reaper exploits originate from security issues a developer has unintentionally left in a device. Even if an attacker is not keen on acquiring new knowledge and is not bothered by the inexpugnable desire to examine an IoT device, the vulnerabilities are still easily exploitable. The exploitation of a chain consisting of several non-critical vulnerabilities may lead to a compromise of a device on the whole.

The overwhelming majority of the vulnerabilities are those common for web applications and web resources. The vulnerabilities of the kind are the first to be looked for by researchers of IoT devices, be it bug bounty hunters or security analysts.

While web resource owners can easily and quickly fix a vulnerability and update their resources, it is not that simple when it comes to IoT devices. Even if a developer releases a designated patch, IoT devices users will surely keep their devices outdated for some time.

Reaper exploits a wide range of vulnerabilities. Some of them were disclosed to the public back in 2013:

- Vulnerabilities in Linksys E1500/E2500 (<http://www.s3cur1ty.de/m1adv2013-004>)
- Vulnerabilities in D-Link DIR-600 and DIR-300 (rev B) (<http://www.s3cur1ty.de/m1adv2013-003>)

```

[0x0000d330 25% 300 reaper/95b448bdf6b6c97a33e1d1dbe41678eb]> ?0;f tmp;s.. @ fcn.0000d150+480
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0x00000000 ffff ffff ffff ffff ffff ffff ffff ffff .....
0x00000010 ffff ffff ffff ffff ffff ffff ffff ffff .....
0x00000020 ffff ffff ffff ffff ffff ffff ffff ffff .....
0x00000030 ffff ffff ffff ffff ffff ffff ffff ffff .....
sb 0x00000000 sl 0x00000000 fp 0x00000000
ip 0x00000000 sp 0x00000000 lr 0x00000000
pc 0x00008190 r0 0x00000000 r1 0x00000000
r2 0x00000000 r3 0x00000000 r4 0x00000000
r5 0x00000000 r6 0x00000000 r7 0x00000000
r8 0x00000000 r9 0x00000000 r10 0x00000000
r11 0x00000000 r12 0x00000000 r13 0x00000000
r14 0x00000000 r15 0x00008190 cpsr 0x00000000
blank 0x00000000
: 0x0000d330 ea1900eb bl fcn.00013ae0 ;[1]
: 0x0000d334 0130a0e3 mov r3, 1 ; 1
: 0x0000d338 d5ffffea b 0xd294 ;[2]
: ; JMP XREF from 0x0000d2e8 (fcn.0000d150)
: 0x0000d33c 0700a0e1 mov r0, r7
: 0x0000d340 dc109fe5 ldr r1, aav.0x00019d34 ; [0xd424:4]=0x19d34 aav.0x00019d34
: 0x0000d344 ac1a00eb bl fcn.00013dfc ;[3]
: 0x0000d348 000050e3 cmp r0, 0
: 0x0000d34c 0400000a beq 0xd364 ;[4]
: 0x0000d350 0900a0e1 mov r0, sb
: 0x0000d354 cc109fe5 ldr r1, str.dlink850l ; [0xd428:4]=0x19d38 str.dlink850l
: 0x0000d358 150d00eb bl fcn.000107b4 ;[5]
: 0x0000d35c 0130a0e3 mov r3, 1 ; 1
: 0x0000d360 cbffffea b 0xd294 ;[2]
: ; JMP XREF from 0x0000d34c (fcn.0000d150)
: 0x0000d364 0700a0e1 mov r0, r7
: 0x0000d368 bc109fe5 ldr r1, aav.0x00019d44 ; [0xd42c:4]=0x19d44 aav.0x00019d44
: 0x0000d36c a21a00eb bl fcn.00013dfc ;[3]
: 0x0000d370 000050e3 cmp r0, 0
: 0x0000d374 0400000a beq 0xd38c ;[6]
: 0x0000d378 0900a0e1 mov r0, sb
: 0x0000d37c ac109fe5 ldr r1, str.dlink860l ; [0xd430:4]=0x19d48 str.dlink860l
: 0x0000d380 0b0d00eb bl fcn.000107b4 ;[5]
: 0x0000d384 0130a0e3 mov r3, 1 ; 1
: 0x0000d388 c1ffffea b 0xd294 ;[2]
: ; JMP XREF from 0x0000d374 (fcn.0000d150)
: 0x0000d38c 0700a0e1 mov r0, r7
: 0x0000d390 9c109fe5 ldr r1, aav.0x00019d54 ; [0xd434:4]=0x19d54 aav.0x00019d54
: 0x0000d394 981a00eb bl fcn.00013dfc ;[3]
: 0x0000d398 000050e3 cmp r0, 0
: 0x0000d39c 0400000a beq 0xd3b4 ;[7]
: 0x0000d3a0 0900a0e1 mov r0, sb
: 0x0000d3a4 8c109fe5 ldr r1, str.dlink865l ; [0xd438:4]=0x19d58 str.dlink865l
: 0x0000d3a8 010d00eb bl fcn.000107b4 ;[5]
: 0x0000d3ac 0130a0e3 mov r3, 1 ; 1
: 0x0000d3b0 b7ffffea b 0xd294 ;[2]
: ; JMP XREF from 0x0000d39c (fcn.0000d150)
: 0x0000d3b4 0700a0e1 mov r0, r7
: 0x0000d3b8 7c109fe5 ldr r1, aav.0x00019d64 ; [0xd43c:4]=0x19d64 aav.0x00019d64
: 0x0000d3bc 8e1a00eb bl fcn.00013dfc ;[3]
: 0x0000d3c0 000050e3 cmp r0, 0
: 0x0000d3c4 0400000a beq 0xd3dc ;[8]
: 0x0000d3c8 0900a0e1 mov r0, sb
: 0x0000d3cc 6c109fe5 ldr r1, str.dlink868l ; [0xd440:4]=0x19d68 str.dlink868l
: 0x0000d3d0 f70c00eb bl fcn.000107b4 ;[5]

```

### *Piece of code to check the model of D-link routers (95b448bdf6b6c97a33e1d1dbe41678eb)*

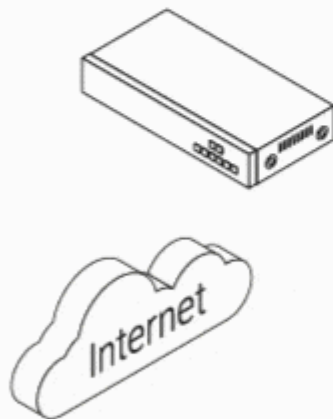
Most of them have been already closed by developers. The exploitation is still possible, though, while the firmware of many devices has not been updated.

Lately, it was also found out that Reaper can also exploit the vulnerabilities of Wireless IP Camera (P2P) WIFICAM detected in March 2017.

## DVR

### Vulnerabilities

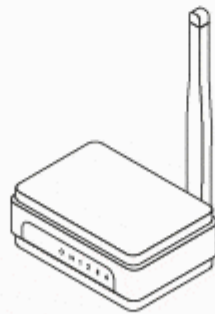
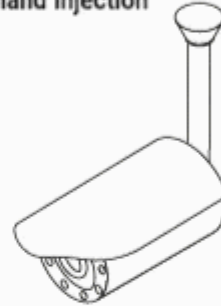
- Weak Password Requirements
- Reliance on Cookies without Validation and Integrity Checking
- Improper Access Control
- Command Shell in Externally Accessible Directory



## Camera

### Vulnerabilities

- Weak Password Requirements
- Cross-Site Request Forgery
- Information Exposure
- Improper Input Validation
- Permissions, Privileges, and Access Controls
- OS Command Injection

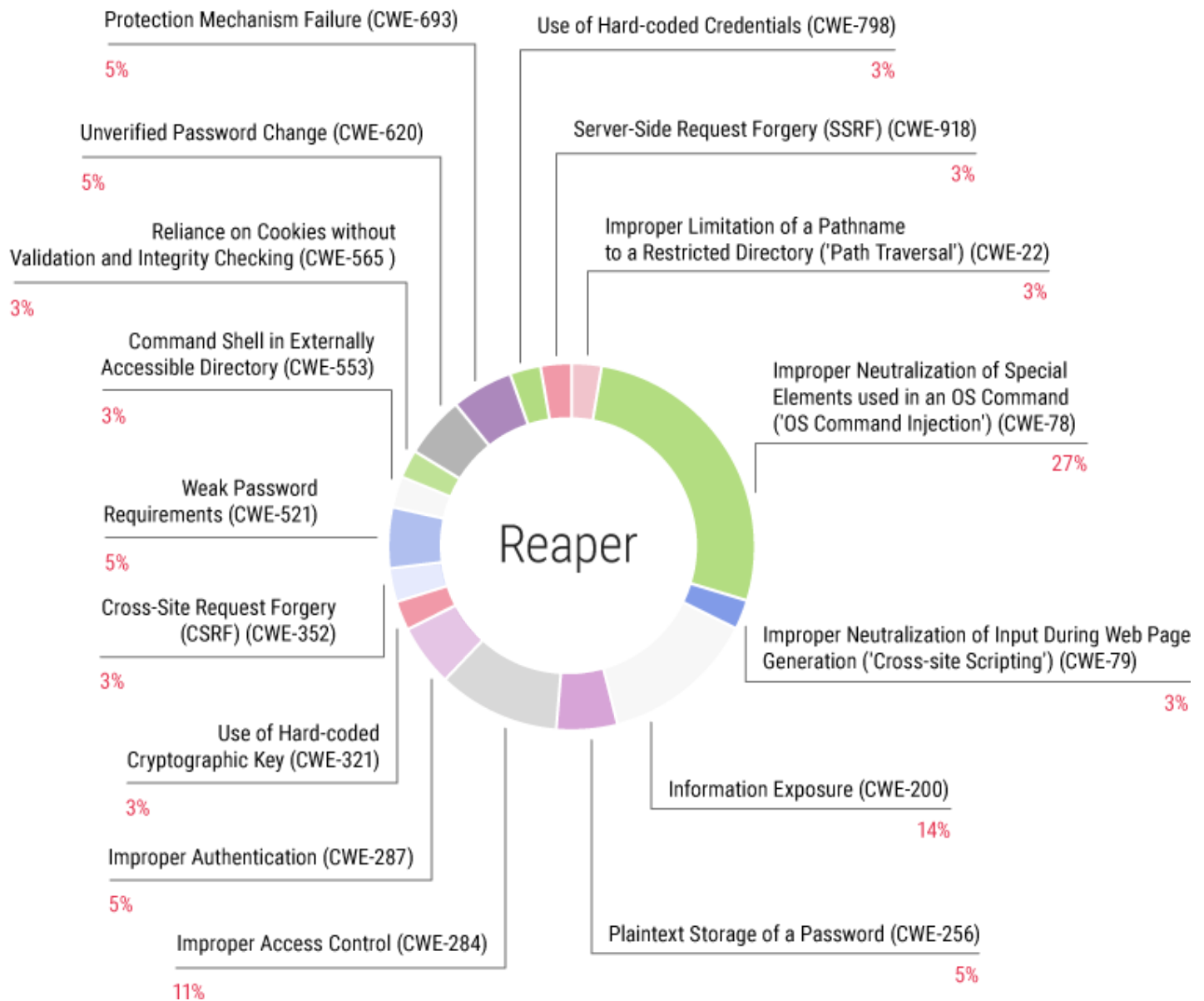


## Router

### Vulnerabilities

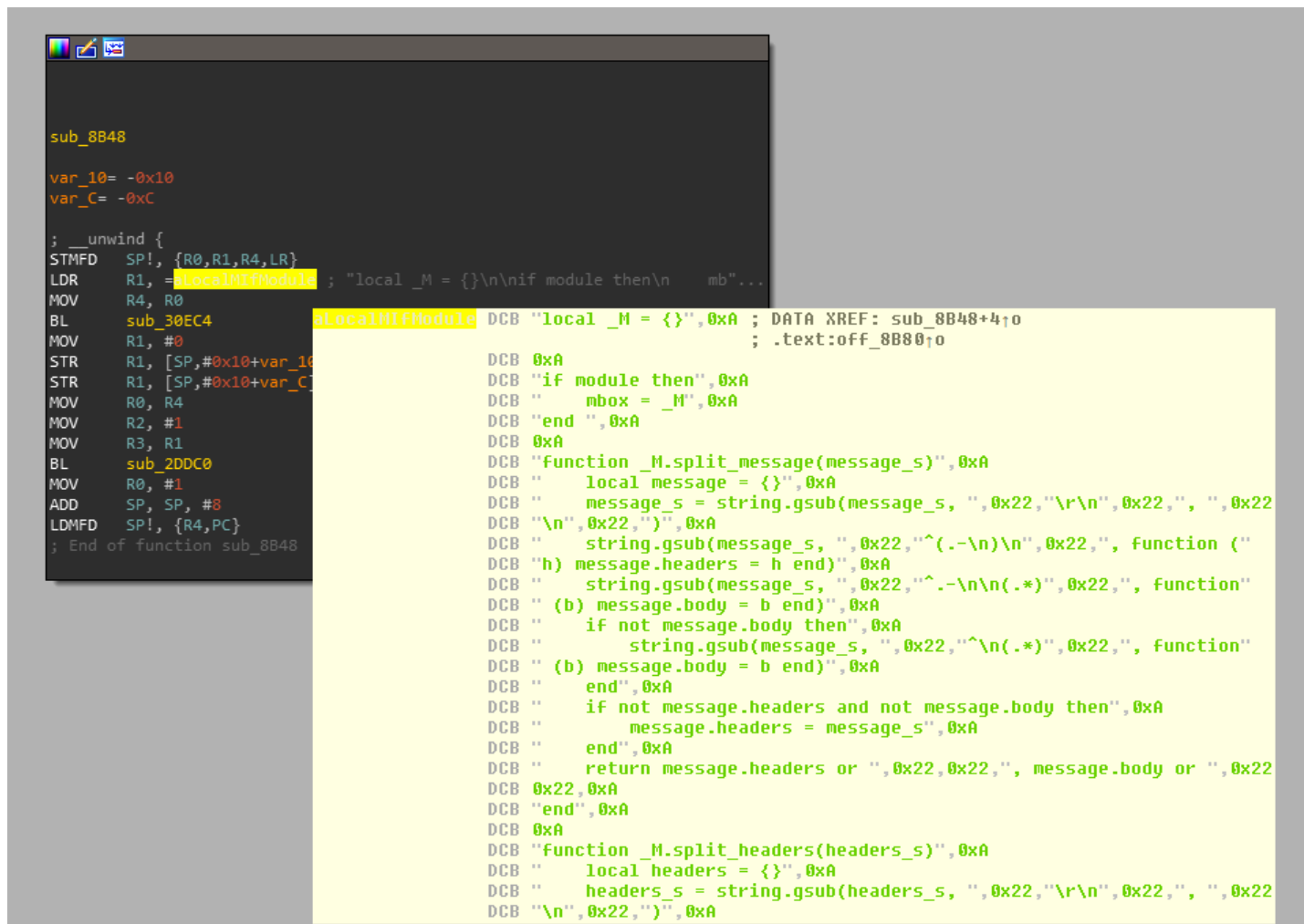
- Improper Input Validation
- Weak Password Requirements
- Information Exposure
- Cross-site Scripting

In the scope of the Embedi's research, the vulnerabilities were categorized by the CWEs assigned to them. The chart below shows the correlation among those devices containing vulnerabilities of the same type.



According to the types of the vulnerabilities on the chart, it is quite obvious that all of them belong to application security. In other words, the used Security Development Lifecycle measures are either insufficient or not implemented at all, which leads to botnets springing up like mushrooms. Moreover, the issues of application security cannot be coped with even if developers “burden” themselves with developing reliable trusted boot and multiple encryption mechanisms.

These vulnerabilities are rare occurrence as long as desktop software is considered and are not easily exploitable due to various OS security mechanisms. The reason for this is plain and clear: as a rule, IoT device developers are not eager to learn from their colleagues’ experience. As for malicious software developers...well, the situation is quite the opposite. For example, Reaper has been equipped with Lua interpreter, which enables more rapid expanding of the functionality of this malicious software.

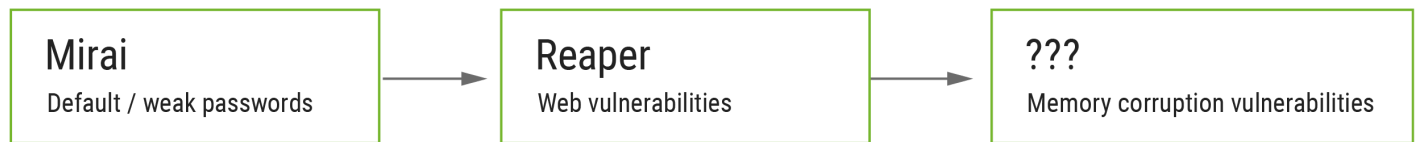


### Example of Lua script for SMPT from Reaper sample (ca92a3b74a65ce06035fcc280740daf6)

Another example is Satori – a botnet based on the source code of Mirai. The main difference between the botnets is that Satori uses a modified method of the distribution and addition of the exploit for a 0-day vulnerability in the Huawei HG532 routers. As The Hacker News states (<https://thehackernews.com/2017/12/satori-mirai-iot-botnet.html>), about 200,000 devices were infected. It also should be noted, that both the exploit and the Mirai source code are publicly available.

Considering this, it is quite obvious that to create such a botnet one needs either to find a 0-day vulnerability in a widely-spread device or to add an exploit for a 1-day vulnerability. Due to application security of most devices being extremely poor, it is as easy as an apple pie for an attacker to find new 0-days and add them to already-existing botnets. If a developer does not improve its application security, the news about hundreds of thousand of devices being infected will hit the headlines with striking regularity.

Taking the expatiation above into account, one can easily map the further development of botnets for IoT devices:



Another little reminder for you: initially Mirai was intended to DDoS the Minecraft servers of its creators' competitors. One can only guess, what will happen if the botnets of the kind are developed by more skillful and experienced teams.

How can end-users ensure their security?

- Purchase the devices capable to automatically update their firmware;
- Regularly update the firmware of their devices;
- Keep up to date with new IoT security threats.

How can developers ensure the security of their devices?

- Develop application security;
- Promptly fix vulnerabilities in their devices and release security updates;
- Automate updates installation on their devices or make the installation process as simple as possible for end-users.

Vulnerabilities exploitable by IoT Reaper

1. D-Link 850L vulnerabilities:

- Remote Code Execution (RCE) via WAN and LAN (CWE-284 (<https://cwe.mitre.org/data/definitions/284.html>))
- Remote unauthenticated information exposure via WAN and LAN (CWE-200 (<https://cwe.mitre.org/data/definitions/200.html>))
- Remote unauthorized program run as administrator via LAN (CWE-284 (<https://cwe.mitre.org/data/definitions/284.html>))

2. Wireless IP Camera (P2P) WIFICAM vulnerabilities:

- Access to administrator account via telnetd (CWE-798 (<https://cwe.mitre.org/data/definitions/798.html>))
- RSA certificate and private key are store in firmware (CWE-321 (<https://cwe.mitre.org/data/definitions/321.html>))
- Remote Code Execution (RCE) with administrator privileges (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))
- Vulnerability of an HTTP server enabling an attacker to bypass verification by sending an empty sign-on parameter to a system and empty loginpas parameter to URL (CWE-693 (<https://cwe.mitre.org/data/definitions/693.html>))
- HTTP server vulnerability in combination with RCE exploitation as an administrator provide an attacker with an opportunity to get an RCE before an administrator account is unauthenticated in a local network



or the Internet (CWE-284 (<https://cwe.mitre.org/data/definitions/284.html>), CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))

- Data streaming monitoring without verification with the help of RTSP server (CWE-287 (<https://cwe.mitre.org/data/definitions/287.html>))
- Traffic monitoring with the help of the tcpdump tool (CWE-200 (<https://cwe.mitre.org/data/definitions/200.html>))

### 3. DVR vulnerabilities:

- Logins and passwords are set by default. Username – admin, password is blank (CWE-521 (<https://cwe.mitre.org/data/definitions/521.html>))
- No cookie content validation allows an attacker to bypass web authentication (CWE-565 (<https://cwe.mitre.org/data/definitions/565.html>))
- An attacker can open the uboot console and set a device into the single-user mode, thus getting an opportunity to execute any command without being authenticated (CWE-284 (<https://cwe.mitre.org/data/definitions/284.html>))
- By using a built-in command prompt shell, an unauthenticated attacker can start telnet (CWE-553 (<https://cwe.mitre.org/data/definitions/553.html>))

### 4. Netgear router vulnerabilities:

- Remote Code Execution (RCE) with the help of command prompt without user authentication (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))

### 5. Vacon surveillance cameras vulnerabilities:

- There is no filtering of input command in board.cgi. It is possible to run cmd and execute any command (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))

### 6. Netgear DGN1000 and DGN2200 v1 routers vulnerabilities

- Built-in web server skips verification of those URLs containing the «currentsetting.htm» line. The vulnerability can be exploited to bypass verification mechanism and to execute arbitrary commands with administrator privileges on devices (CWE-20 (<https://cwe.mitre.org/data/definitions/20.html>))

### 7. Linksys E1500/E2500 routers vulnerabilities:

- The ping\_size parameter is not checked. An attacker can input and execute arbitrary shell commands (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))
- An attacker can change a current password without knowing it (CWE-620 (<https://cwe.mitre.org/data/definitions/620.html>))

### 8. D-Link DIR-600 and DIR-300 vulnerabilities

- No access restrictions and no input verification in command prompt allow an attacker to input and execute arbitrary commands. For example, to run telnetd (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))
- An attacker can change a set password even if it is unknown (CWE-620 (<https://cwe.mitre.org/data/definitions/620.html>))
- Passwords are not hashed and stored in plain text (CWE-256 (<https://cwe.mitre.org/data/definitions/256.html>))

- Unauthenticated access to information about a device model name, firmware version, language and MAC addresses, and Linux kernel (CWE-200 (<https://cwe.mitre.org/data/definitions/200.html>))
- Access to information about an OS and its location (CWE-200 (<https://cwe.mitre.org/data/definitions/200.html>))
- There is no verification of input commands in the SSID parameter. Input code is run if an attacker attempts to access the version.txt file. Authentication is not required (CWE-79 (<https://cwe.mitre.org/data/definitions/79.html>))

#### 9. Avtech surveillance cameras vulnerabilities:

- Passwords are stored in a text file. If there is access to a device, there is access to passwords as well (CWE-256 (<https://cwe.mitre.org/data/definitions/256.html>))
- Web interface is not protected from CSRF attacks (CWE-352 (<https://cwe.mitre.org/data/definitions/352.html>))
- CGI script can be obtained in the “/cgi-bin/nobody” folder without authentication (CWE-200 (<https://cwe.mitre.org/data/definitions/200.html>))
- Unauthorized SSRF on DVR devices (grants access to IP cameras connected to a local network). By modifying parameters ip, port, and queryb64str, and attacker can execute arbitrary HTTP requests via a DVR device without being authenticated (CWE-918 (<https://cwe.mitre.org/data/definitions/918.html>))
- Unauthorized command input on DVR devices. Search.cgi does not check received parameters, which can be used by an attacker to conduct RCE (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))
- Authentication bypass by adding the “.cab” or “/nobody” lines in an URL request (CWE-287 (<https://cwe.mitre.org/data/definitions/287.html>))
- Unauthorized loading of a file from the root of a catalog (CWE-22 (<https://cwe.mitre.org/data/definitions/22.html>))
- CAPTCHA bypass by inputting the “login=quick” parameter (CWE-693 (<https://cwe.mitre.org/data/definitions/693.html>))
- The exeFile parameter is not checked. Consequently, an attacker can execute arbitrary system commands with administrator privileges (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))
- The DoShellCmd function parameters are not checked. An attacker can execute arbitrary system commands with administrator privileges (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))
- PwdGrp.cgi uses the parameters of a username, password, and a group in a new request to create or modify a user in a system command without checking (CWE-78 (<https://cwe.mitre.org/data/definitions/78.html>))

Download infographic (/wp-content/uploads/2018/01/Reaper-Pics.png)

P.S. We were glad to find out that our fellow researchers share same views on the problem. Their research was focused on the Satori botnet. So, if you have a moment, please, check their research (<https://researchcenter.paloaltonetworks.com/2018/01/unit42-iot-malware-evolves-harvest-bots-exploiting-zero-day-home-router-vulnerability/>). It is absolutely worth your attention.



(https://www.linkedin.com/shareArticle?mini=true&url=http://embedi.com/blog/grim-  
 iot- iot-  
 reaper- reaper-  
 1- 1-  
 and- and-  
 0- 0-  
 day- day-  
 vulnerabilities-  
 at- at-  
 the- the-  
 service-service-  
 of- of-  
 botnets/botnets&context=Grim  
 IoT IoT  
 Reaper:Reaper:  
 1- 1-  
 and and  
 0- 0-  
 day day  
 vulnerabilities  
 at at  
 the the  
 service service  
 of of  
 botnets)botnets)



( ) 30

botnets)botnets)

Subscribe to our newsletter to stay in touch

Enter your email here



Submit


Solutions (<https://embedi.com/solutions/>) Blog (<https://embedi.com/blog/>)

Our news (<https://embedi.com/news/>) Resources (<https://embedi.com/resources/>)

About (<https://embedi.com/about/>)

**f** (<https://www.facebook.com/Embedi/>) **in** (<https://www.linkedin.com/company/embedi>)

 ([https://twitter.com/\\_embedi\\_](https://twitter.com/_embedi_))  (<https://github.com/embedi/>)

 (<https://www.youtube.com/channel/UC9XR2noZynNxvQcg0G9ooKA>)

2016 - 2018 ©