Through the Looking-Glass, and what Eve found there

http://www.s3.eurecom.fr/lg/

Luca 'kaeso' Bruno <lucab@debian.org>,
Mariano 'emdel' Graziano <graziano@eurecom.fr>
About us

• S3 group at Eurecom (FR) - **System security**
  – Embedded systems
  – Networking devices
  – Critical infrastructures
  – Memory forensics
  – Malware research
Outline

• **Motivations**
• Intro to looking glasses
• Threats
• Vulns & incidents
• Countermeasures
Motivations – how this started

• Picture yourself as a newbie cyber-criminal looking for the next target
  – Aim: critical infrastructure
  – Impact: worldwide
  – Skill level: low
  – Goal: break havoc
Motivations – how this started

• Picture yourself as a newbie cyber-criminal looking for the next target
  – The Internet
  – Impact: worldwide
  – Skill level: low
  – Goal: break havoc
Motivations – how this started

- Picture yourself as a newbie cyber-criminal looking for the next target
  - The Internet
  - Traffic routing across ASes
  - Skill level: low
  - Goal: break havoc
Motivations – how this started

• Picture yourself as a newbie cyber-criminal looking for the next target
  – The Internet
  – Traffic routing across ASes
  – Basic web skills, google dorks, etc...
  – Goal: break havoc
Motivations – how this started

• Picture yourself as a newbie cyber-criminal looking for the next target
  – The **Internet**
  – Traffic **routing across ASes**
  – **Basic web skills**, google dorks, etc...
  – Gaining access to **BGP routers**
Motivations – how this started

• Picture yourself as a newbie cyber-criminal looking for the next target

A good candidate: LOOKING-GLASS
Outline

• Motivations
• Intro to looking glasses
• Threats
• Vulns & incidents
• Countermeasures
The Internet

• A network of networks, glued by BGP

http://www.caida.org/research/topology/as_core_network/2014/
One routing-table, many routing-tables

- BGP is worldwide, each AS routing table is a (partial) local view
- What you see depends on where you are

http://blog.thousandeyes.com/4-real-bgp-troubleshooting-scenarios/
Connectivity troubleshooting

• NOC tools for troubleshooting:
  – Distributed BGP probes, eg. RIPE Labs
  – Private shells exchange, eg. NLNOG
  – Limited web-access to routers, ie. via looking-glasses
What's in a looking glass

• A simple '90s style web-script:
  – Usually PHP or Perl
  – Single file, can be dropped in webroot
  – Direct connection to SSH/telnet router console
  – Cleartext config file (ie. credentials)
How does it work

Public IP (data+BGP)
Private admin (telnet/SSH)
Public web (looking-glass)
Public IP (data+BGP)
How does it look like

Looking Glass

<table>
<thead>
<tr>
<th>Type of Query</th>
<th>Additional parameters</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bgp advertised-routes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bgp summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trace</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Disclaimer: All commands will be logged for possible later analysis and statistics. If you don't like this policy, please disconnect now!

Please email questions or comments to

10/08/2014
Where to get it

- Focus on open-source most common ones:
  - Cougar LG (Perl)
  - Cistron LG (Perl)
  - MRLG (Perl)
  - MRLG4PHP (PHP)
Outline

• Motivations
• Intro to looking glasses
• Threats
• Vulns & incidents
• Countermeasures
Targeting humans

- Assume **bug-proof** software
- Humans can still mis-deploy it, and forget to:
  - Enable CGI/mod_php/mod_perl
  - Protect config files
  - Protect private SSH keys

Exposed routers credentials
Targeting the web-app

• Assume some minor bugs may exist in the web frontend

• Pwn the LG web interface:
  – Improper escaping
  – XSS/CSRF/etc.

Cookie stealing for other web services
Targeting the server

• Assume some medium severity bugs may exist in the whole package

• Pwn the host through LG:
  – Embedded third-party tools
  – Forked/modified modules

Escalate to the hosting server
Targeting the router

• Assume **important** bugs may exist in the backend

• Pwn the router through LG:
  - Missing input escaping
  - Command injection to router
  - Known bugs in router CLI

  Escalate to router administration
Targeting the Internet

• Assume you control multiple routers in multiple ASes

• Pwn the Internet:
  – Reroute/blackhole local traffic
  – Announce bogus BGP prefixes

Chaos ensues :)
Outline

- Motivations
- Intro to looking glasses
- Threat model
- Vulns & incidents
- Countermeasures
Web issues

• Exposed Credentials:
  – Stored in cleartext: IPs, usernames and passwords
  – Configuration files at known URLs

• Cookie Stealing:
  – XSS vulnerabilities in LG, to target other web-apps
Web Misconfigurations

• Google Dorks for login credentials:
  – Find LG configuration files
  – Examples:
    • "login" "telnet" inurl:lg.conf
    • "login" "pass" inurl:lg.cfg
Google Dorks – Exposing conf files

```
inurl:lg.conf "telnet"
```

5 risultati (0.16 secondi)

**lg.conf(5)**

and programs needed within these, such as `telnet(1)`, are located. Its value is set by configure. Should it be necessary to modify PATH, note that it must include ...

**lg.log ./as.db ../logo.gif Looking Glass favicon ...**

... analysis and statistics. If you don't like this policy, please disconnect now! On telnet://... telnet://...
Google Dorks – Exposing conf files

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- $Id: lg.conf,v 1.9 2004/01/25 20:19:45 cougar Exp $ -->

<LG_Conf_File>

  <LGURL> </LGURL>
  <LogFile>lg.log</LogFile>
  <ASList>/as db</ASList>
  <LogoImage Align="center" Link="http://www.lookup.com">..../Looking Glass</LogoImage>
  <Favicon>favicon.ico</Favicon>
  <ContactMail>backbone@server.com</ContactMail>
  <RSHCmd>/usr/bin/rsh -l lg</RSHCmd>
  <HTTPMethod>POST</HTTPMethod>  <!-- use "GET" if you like to
  <Timeout>25</Timeout>
  <Disclaimer>All commands will be logged for possible later analys
  <SecureMode>On</SecureMode>

  <Router_List>

    <!-- (AS*) Looking Glass -->

    <Router Name="">
      <URL>telnet://\192.168.1.20.1</URL>
    </Router>

    <Router Name="">
      <URL>telnet://\192.168.1.19.254</URL>
    </Router>

  </Router_List>

</LG_Conf_File>
Default config paths

● Example from Cougar LG root directory:

as.txt  CHANGELOG  communities.txt  COPYING  favicon.ico
lg.cgi  lg.conf  makeaslist.pl  makedb.pl  README

● So just crawl for it:

$BASE_LG_URL/lg.conf
Best Practices :) 

README sometime mentions them:

```
21   Then copy the lg.pl, lg.cfg and lg.html.inc files to a subdirectory on
22   your webserver. Make sure that those files are readable by your webserver,
23   and that lg.pl is also executable. Make sure there is NO WORLD READ ACCESS
24   on the lg.cfg file since it contains YOUR CISCO PASSWORD (hope you get it).
25
26   Because your Cisco password is in the configuration file, it is preferable
27   to run this script on a web server where no one else has access to - not
28   the virtualhosting server for all your customers...
29```

...still, we've found about 35 exposed cases!
Exposed Source Code

In use LG

(Hopefully) non-working LG
Exposed Private SSH Keys

• Default path for SSH keys (CVE-2014-3929) in Cougar LG

• Where are SSH private keys stored?
  
  lg.conf:18 → /var/www/.ssh/private_key
Exposed Private SSH Keys

```
-----BEGIN DSA PRIVATE KEY-----
BuwIBAAAABgQDC72pilmrjWYXs8hJqyjyu3VvOZqfMuQLB10A+
leZrelXi1Polji0+imvt9+gMZnzZcmdg1jK+Fq+WRNWCErTmi0aaVG91DwIvANpR
inNVUF2ZG3ah9UcIVcF7Rj8jc3j80UC6wleooO6hkBqbjveRwkJ4Vya8qKo3wLYDw
kiTsM2kCgYBermXmdvZDPT6SO2fUJViixKV+UJkJ9wWddgbnBVRFul2H6CLWHP3x
p9OG/XmFpHjK5B958t88W4RajplglFrO3LgtoK6
j1RCnRcE5YpU5C1q6jyBS+pySDoEmMCjztDX28g2QYxkh1
-----END DSA PRIVATE KEY-----
```

### Index of /lg/.ssh

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍀 Parent Directory</td>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>🎯 id_dsa</td>
<td>03-Jul-2008 11:11</td>
<td>668</td>
<td></td>
</tr>
<tr>
<td>🎯 id_dsa.pub</td>
<td>03-Jul-2008 11:11</td>
<td>615</td>
<td></td>
</tr>
<tr>
<td>🎯 ssh_config</td>
<td>03-Jul-2008 11:11</td>
<td>1.2K</td>
<td></td>
</tr>
</tbody>
</table>

Apache/2.2.14 (Ubuntu) Server at [redacted] Port 80
First steps into the web

• No CAPTCHA anywhere!
• This eases attacker's work:
  – Automated resource mapping (ping-back and conf dumping)
  – Automated command injection
  – **Automated attacks from multiple AS** (if bugs are found)
XSS

- XSS in `<title>` via "addr" parameter (CVE-2014-3926)

- LG maybe are not worthy web targets...
  - But other NOC services often are under the same-origin domain!
XSS – for the lulz!

Looking Glass

<table>
<thead>
<tr>
<th>Type of Query</th>
<th>Additional parameters</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>bgp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bgp advertised-routes</td>
<td>8.8.8.8&lt;/TITLE&gt;&lt;/head&gt;&lt;body&gt;&lt;sc</td>
<td>EDGE1-TC1</td>
</tr>
<tr>
<td>bgp summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>trace</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Submit  | Reset

The page at lg.

hello NOC :)

OK
Router Command Injection

• What if you can run whatever CLI command you want?
  – **CVE-2014-3927** in MRLG4PHP
• 'argument' parameter issue
  – HTML escape != sanitization
• Let's look at the code (mrlg-lib.php:120)
Router Command Injection

```php
else $argument = trim ($_REQUEST["argument"]);

$command = $_request[$_requestid]["command"] . (!empty ($argument) ? " " . safeOutput ($argument)) : "");

global $socket_timeout;
$link = fsockopen ($address, $port, $errno, $errstr, $socket_timeout);
if (!$link)
{
    printError ("Error connecting to router");
    return;
}

socket_set_timeout ($link, $socket_timeout);
$username = $_router[$_routerid]["username"];\n
if (!empty ($username)) fputs ($link, "$username\n");

fputs ($link, "{$password}\terminal length 0\n{$command}\n");

// let daemon print bulk of records uninterrupted
if (empty ($argument) & $request[$_requestid]["argc"] > 0) sleep (2);

fputs ($link, "quit\n");

function safeOutput ($string)
{
    return htmlentities (substr ($string, 0, 50));
}
```
Router Command Injection - PoC

• From HTTP to router CLI, just adding newlines :)

```
curl --data "
'routerid=10
&requestid=50
&argument=8.8.8.8%0Adate%0Aexit%0A'"
```
Remote Memory Corruption

- Sometime LG ships with embedded third-party binaries
  - **CVE-2014-3931** in MRLG (fastping SUID bin)
- ICMP echo reply is used without proper validation
  - fastping.c:546
    Riempie_Ritardi( *((long *)&(icp->icmp_data[8])) , triptime );
- Let's have a look at the code
Remote Memory Corruption

```c
// ### Stampa il ritardo del pacchetto ricevuto
Riempi_Ritardi( *((long *)&(icp->icmp_data[8])) , triptime );

870 /* Inserisce nel vettore ritardi il RTD di ogni pacchetto */
871 void Riempi_Ritardi ( long indice, long ritardo )
872 {
873     /* controllo se è presente un fuori sequenza */
874     if ( indice < prec )
875     {
876         fuoriseq++;
877         //printf("%ld\n",fuoriseq);
878     }
879
880
881     prec=indice;
882     //printf("%ld\n",indice);
883     ritardi[indice]=ritardo;
884     }
885 }
```
Exploitation notes

• 3rd-party, probably not commonly deployed
  – WONTFIX by upstream
• Time-dependent...
  – But you get host time in ICMP echo request!
• Every ICMP reply can overwrite one long word in memory...
  – And you have 100 probes on every try
Talking about network design

- Routers admin consoles needlessly exposed over **globally routable** interfaces
Outline

- Motivations
- Intro to looking glasses
- Threat model
- Vulns & incidents
- Countermeasures
Code-wise

• Understand that exposing router consoles to the web with hardcoded credentials can be dangerous!

• Review all critical web-services written during the wild-west '90s
Deployment-wise

• Prefer a dedicated read-only route-server as LG endpoint

• Check if your private files are reachable over the web (LG config, SSH keys)

• **Double check your web server config!** (vhost vs. default docroot)
Administration-wise

- Setup proper ACL on your routers
- Use strong, unique passwords
- Put admin and out-of-band services in private VLANs and subnets!
Recap

- Best-practices are often disregarded
- Unaudited, old, forgotten code often sits in critical places
- Attackers go for the weak links...
  - and escalate quickly!

Internet core is fragile
Fin

Thank you for listening!

Thanks to all the members of NOPS team, who helped in bug-finding
Backup – router CLI escalation

• Cracking Cisco weak hashes
  – Type-0, Type-5, Type-4 (cisco-sr-20130318-type4)

• Exploiting CLI bugs
  – Cisco, AAA Command Authorization by-pass (cisco-sr-20060125-aaatcl)
  – Juniper, Unauthorized user can obtain root access using CLI (JSA10420)
  – Juniper, Multiple privilege escalation vulnerabilities in Junos CLI (JSA10608)
Backup – reported incidents

<table>
<thead>
<tr>
<th>Vulnerabilities</th>
<th>Affected ASes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exposed configuration files</td>
<td>28</td>
</tr>
<tr>
<td>Remote command injection</td>
<td>12</td>
</tr>
<tr>
<td>Misconfigured CGI</td>
<td>4</td>
</tr>
<tr>
<td>Exposed SSH private keys</td>
<td>2</td>
</tr>
</tbody>
</table>