High Security Laboratory - Network Telescope Infrastructure Upgrade

Frédéric Beck, Alexandre Boeglin, Olivier Festor

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

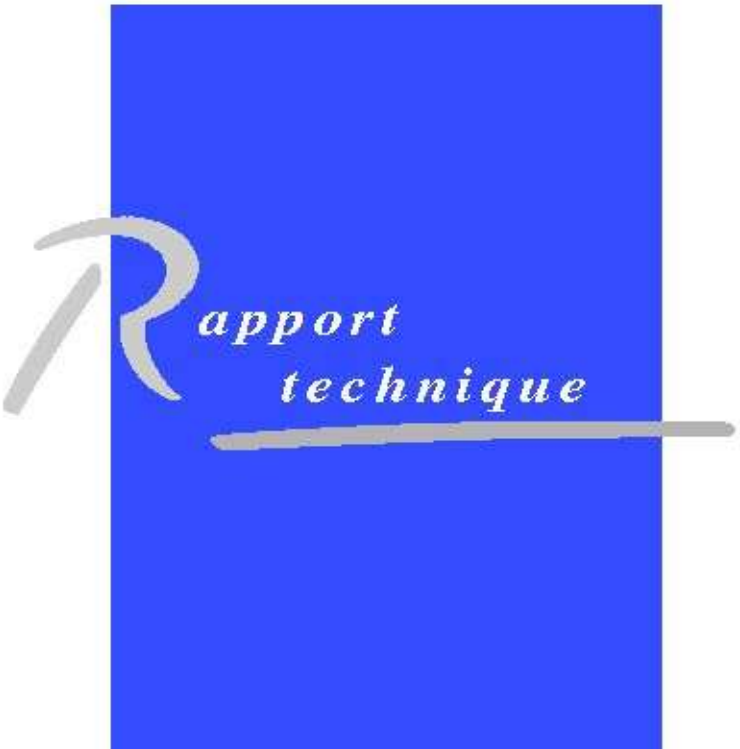
High Security Laboratory - Network Telescope Infrastructure Upgrade

Frédéric Beck, Alexandre Boeglin and Olivier Festor

N° 9999

March 2007

Thème COM



ISSN 0249-0803

High Security Laboratory - Network Telescope Infrastructure Upgrade

Frédéric Beck, Alexandre Boeglin and Olivier Festor

Thème COM — Systèmes communicants
Projet MADYNES

Rapport technique n° 9999 — March 2007 — 17 pages

Abstract:

Key-words: security, network, telescope, malware

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,

Laboratoire de Haute Sécurité en Informatique - Téléscope Réseau

Résumé :

Mots-clés : sécurité, réseau, téléscope, malware

Contents

Introduction

As part of the High Security Laboratory at INRIA Nancy Grand Est inaugurated in July 2010, we have been running and maintaining a network telescope for more than 2 years. Many updates and upgrades of the different components have been made during this period, as well as the apparition of new threats and vulnerabilities, motivating an upgrade of the existing infrastructure to maintain it up-to-date with the current security issues.

This report is a follow up of the previous report written in May 2008 describing the specification and deployment of the initial infrastructure. In this report, we present the upgrade performed during the second half of the year 2010, after the inauguration and moving of the platform.

1 Physical Infrastructure

1.1 Dedicated room

A new room dedicated to the High Security Laboratory (LHS) has been built in the basement. It is composed of a servers room, one open space, and one room holding the security and access control terminal.

Enhanced physical security and strict access control have been implemented. All doors and windows are armored and bulletproof, and an alarm system is monitoring all accesses or break-ins. In the servers room, a presence and movement sensor is detecting any unwanted entrance. Finally, a noise sensor raises an alarm if the noise level threshold is exceeded (e.g. if one drops a front panel or try to move one server). To prevent false-positive alarms, it is recommended to disable the alarm in the servers room before doing any maintenance task, and reset it when exiting the room.

To access any of these rooms, one must go through an airlock. To pass the first door, one needs to present a smart card to the card reader and one finger to the vein reader. Then, once the main door is shut, using the smart card, one can gain access to the open space or the servers room. Once in the servers room, one can access the security terminal room with a smart card and a retina control.

This last room holds the security and access control terminal, where one can configure access control rules (add or modify users) and check all logs. This room also contains the switchgear cubicle and has a separated AC split that makes possible to add another rack in the room if required. Access to this room should be strictly limited to authorized staff.

The servers room has been designed to hold 4 42 units racks. At the moment, all 4 slots are used by 2 42U racks for the in-vitro analysis cluster, and 2 24U racks for the network telescope. The racks for the telescope are plugged to 2 separate and redundant electrical circuits. Each rack has 2 Power Distribution Units (PDU) plugged to separate circuits. Each server also has redundant power blocks plugged to separate PDUs and thus separate circuits, ensuring full electrical redundancy.

4 AC cupboards have been placed in the room. 2 of them are plugged on the institute AC system (water-air), and two of them are on a separate air-air system. By default, the ones plugged on the institute system are running. When maintenance operations are performed on the system, make sure to switch on the splitted ones. Defaults constraints of 21°C and 35% of air humidity are set. If the temperature raises over 26°C in the room, the electrical input is switched off to ensure hardware integrity. As far as we know, there are no alarm mechanisms that are used to prevent brutal shutdown of the servers when the black out occurs. This is a point that must be dug and maybe implemented to let the administrators safely shut the servers and devices down before the blackout occurs.

1.2 Hardware changes

As part of the leurrecom.org project [1], we added one 1U server to the telescope. It is a Dell PowerEdge R410 server with Intel Xeon Six-Core L5640 2.26GHz CPU, 24GB RAM, 2 SATA HDD of 500GB each configured in RAID 1.

Details about the installation and the project are described in section 3.4.

1.3 Racks

In this section we present the physical implementation of the servers and devices in the different racks.

[1] `http://www.leurrecom.org/`

Figure 1: Collect Environment (left rack)

Unit	**Device**	**KVM**	**Gb1**	**Gb2**	**Gb3**	**Gb4**
24 23	Cisco 2821 router *torgnol*		Ge0/0 Orange SDSL	Ge0/0 meowth-1		
22						
21	Cisco 2960 switch *meowth*		meowth-9 zubat-13	meowth-10 dialga-Gb1		
20						
19 18	Dell PowerEdge 2950 *mew*	10	meowth-8	arcanine-8		Neufbox
17 16	Dell PowerEdge 2950 *togepi*	9	meowth-7	arcanine-7		
15 14	Dell PowerEdge 2950 *onix*	8	meowth-6	arcanine-6		
13	PDU					
12 11	Dell PowerEdge 2950 *charmander*	7	meowth-5	arcanine-5		
10 9	Dell PowerEdge 2950 *squirtle*	6	meowth-4	arcanine-4		
8 7	Dell PowerEdge 2950 *bulbasaur*	5	meowth-3	arcanine-3		Freebox
6 5	Dell PowerEdge 2950 *psyduck*	4	meowth-2	arcanine-2		Orange ADSL Pro
4						
3	Cisco 3560 switch *arcanine*		arcanine-24 zubat-2			
2	PDU					
1	Dell KVM 2161DS-2					

Figure 2: Experimentation Environment (right rack)

Unit	Device	KVM	Gb1	Gb2
24	Cisco 2960 switch *zubat*		zubat-1 arbok-0	zubat-2 arcanine-24
23				
22 21	Dell PowerEdge 2950 *nidoran*	13	zubat-7	zubat-18
20 19	Dell PowerEdge 2950 *geodude*	12	zubat-6	zubat-17
18 17	Dell PowerEdge 2950 *mankey*	11	zubat-5	zubat-16
16				
15	PDU			
14 13	Dell PowerEdge 2950 *jigglypuff*	3	zubat-5	zubat-15
12				
11 10	Dell PowerEdge 2950 *pikachu*	1	zubat-4	zubat-14
9				
8	Cisco ASA firewall *arbok*		arbok-0 zubat-1	arbok-1 dialga-Gb2
7				
6 5	Dell PowerEdge 2950 *dialga*	2	meowth-10	arbok-1
4				
3 2 1	Dell PowerVault MD1000 + PDU at the back in slot 2	dialga		

All ports for switch *meowth* that are not listed in these tables are bound to VLAN 185 (Internet).

1.4 Network connections

No changes have been made on the 3 ADSL connections. We still have Free, Neuf/SFR and Orange ADSL Pro.

However, we upgraded the bandwidth of the Orange SDSL connection to the maximum allowed by the phone line, which is 2Mb/s, after the global upgrade, as we were reaching the limits of the previous 1Mb/s contract.

2 Operating systems upgrade

2.1 Xen hypervisor and Dom0

We began the Dom0 operating system upgrade with the experimentation environment servers *pikachu*, *jigglypuff* and *geodude*. We upgraded the Debian Linux distribution, as well as the Xen hypervisor and utils to version 4.0. The upgrade went fine and all the VMs are running without problem.

However, one problem appeared on *jigglypuff* and *geodude*. When running Xen 4.0 hypervisor, it is not possible to start the X.org server, as there is a know bug with this X server version and Xen 4.0.

Thus, we did not upgrade any other servers, as X may be required on some of them for some experiments, e.g. running some tools such as wireshark to analyze traffic on bridge xenbr0 in Dom0.

2.2 VMWare

For some of the experiments that may run on the experimentation environment, full hardware virtualization/emulation may be required (e.g. VoIP honeypots or bots that do not work very well with fake/dummy sound cards). To make this possible, we installed a VMWare Player framework on the server *geodude* running Ubuntu 10.04.

This also allows to easily and quickly run operating systems that are more difficult to get running on Xen, such as Windows, BSD or some Linux distributions (e.g. CentOS). Ready to deploy images are available at `http://vmplanet.net/` or `http://vmware.pouf.org/`.

In order to enable a better control of these VMs, an upgrade of the solution to VMWare vSphere Hypervisor can be made [2].

2.3 Virtual machines

After the upgrade, we planned to update all VMs as well. However, due to the bug with X server, we postponed this task. We kept the basis of our VMs (kernel 2.6.18) and simply updated the Debian OS within these VMs, as these guests work fine with both Xen 3.X and 4.0 versions.

We generated a new reference VM based on Debian SID, instantiated and stored on dialga, which we used as reference for the soft upgrade.

[2] `http://www.vmware.com/fr/products/vsphere-hypervisor/`

3 Telescope upgrade

3.1 Surfnet IDS 3.05

The first step was to upgrade the database server itself to match the requirements of Surfnet IDS 3.05.

```
apt-get install postgresql
pg_dropcluster 8.3 main --stop
pg_upgradecluster 8.1 main
```

Then, we installed the logging server as detailed in `http://ids.surfnet.nl/wiki/doku.php?id=latest_docs:1_logging_server:1._installation`.

By following the information at `http://ids.surfnet.nl/wiki/doku.php?id=latest_docs:1_logging_server:2._configuration`, we configured the logserver. It important to no activate the option *c_minifield_enable*, even if the comments state that it will boost the web interface, because many javascripts are missing. Moreover, one must get the Geolite data manually at `http://www.maxmind.com/app/geolitecity` and put the *GeoLiteCity.dat* file manually in */opt/surfnetids/include* as the automatic script does not work well.

We obtained a working logserver with no data loss, supporting new honeypots, and thus allowing more variety in terms of honeypots and vulnerabilities.

3.2 Honeypots

To register the new sensors, we modified the *localsensor.pl* script from the tunnelserver, and created the script *update_sensors.pl*. This script updates the *sensors* table with the new information and adds records in the newly created *sensor_details* table. Thus, we are not creating new sensors but reusing the same identifiers than before, we only update the sensor name, the honeypot type being set when logging attacks.

One important change here is that the sensors are now considered permanent, and the flags in the *Sensors Status* tab in the web interface can not be used anymore to detect sensors failures. To check if a sensor is running, you must now enter the *Sensor Details* tab by clicking on the sensor name, and check the field *Last keepalive*.

3.2.1 Nepenthes

Nepenthes is the sole honeypot that was used in the first deployment of the telescope. The project moved to `http://nepenthes.carnivore.it/`, and is not active anymore, as it has been replaced by Dionaea3.2.2.

We kept 14 instances of Nepenthes running on the server *mew*.

3.2.2 Dionaea

Dionaea [3] is meant to be a nepenthes successor, embedding python as scripting language, using libemu to detect shellcodes, supporting ipv6 and tls.

[3] `http://dionaea.carnivore.it/`

By following the instructions at `http://dionaea.carnivore.it/#compiling`, we installed and configured Dionaea in a new VM called *dionaea-reference*. All utility scripts we had previously written for Nepenthes have been updated to support dionaea:

dionaea-clone.pl allows to clone the VMs

dionaea-alive.pl updates the sensors details table's timestamp to tell the logserver that the sensor is still alive, and restarts the honeypot if required

dionaea-scp.pl using the new upload user and keys, uploads the captured binaries and binary streams to *dialga*

We first deployed instances of Dionaea on all servers in the collect environment (excluding *mew*) and all ADSL connections. However, we were saturating the network connection and downloading 18 000 malwares per day, most of them redundantly, on the sensors. To favor the diversity of the collected data and emulated vulnerabilities, we decided to limit the deployment to ADSL connections (to keep the opportunity to analyze the differences between them) and 12 instances on servers *psyduck*, *bulbasaur* and *squirtle*, on the SDSL connection.

3.2.3 Kippo

Kippo [4] is a medium interaction SSH honeypot designed to log brute force attacks and, most importantly, the entire shell interaction performed by the attacker. Kippo is inspired, but not based on Kojoney.

By basing us on the instructions at `http://ids.surfnet.nl/wiki/doku.php?id=kb:installing_kippo`, we created and configured a new reference VM for kippo called *kippo-reference*. The main differences with the documentation are the addition of a kippo user and group that is used to run kippo as a non-root system user via the init.d script. This user is not allowed to log into the system, its sole role is to run kippo.

As kippo is emulating an SSH server, it tries to bind the port 22. But as it is running as a non-root user it fails. Thus, kippo is configured to bind the port 2222, and the traffic is redirected to this port via Netfilter *REDIRECT* target, that ensures that the source IP is not altered or modified, with the command:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22 -j REDIRECT --to-port 2222
```

Of course, the real SSH server running on the VM is not binding on port 22, but binds the port TCP 2220. Once again, all the utility scripts have been written for this new honeypot. The only difference with the usual ones is that, for stability issues, kippo is restarted every 30 minutes automatically. The logserver's database has been updated to support Kippo, as well as the web interface.

At the moment, kippo does not support non-root logins. However, several passwords have been set:

[4] `http://code.goggle.com/p/kippo/`

- one in the configuration file which is the same than the hostname of the emulated host
- several obvious passwords have been insert in a DBM located at */opt/kippo/data/pass.db* with the tool */opt/kippo/utils/passdb.py*

3.2.4 Amun

Amun [5] is another honeypot similar to Nepenthes and Dionaea. In order to diversify the detection methods and vulnerabilities, we deployed 14 instances on the server *onix* by following the documentation at `http://ids.surfnet.nl/wiki/doku.php?id=latest_docs:2_tunnel_server:1d._amun`.

We wrote all support scripts and configured the honeypot to log into the logserver. The *amun-clone.pl* script also modifies the sensor IP in the configuration file */opt/amunhoney/conf/log-surfnet.conf* file in the VM.

3.3 Traces

3.3.1 TCPDump

No major changes have been made. We are still capturing traffic on all network bridges to the Internet (xenbr0) in xen Dom0, but the upload of the traces is done with the *upload* user.

3.3.2 Netflow

Netflow probes are running on various servers to monitor the flows going through the network bridge to the Internet (xenbr0). The following table presents all the probes:

Figure 3: Netflow probes in the telescope

Source	Destination	Destination Port
psyduck	dialga - 10.1.1.1	9556
bulbasaur	dialga - 10.1.1.1	555
squirtle	dialga - 10.1.1.1	9557
charmander	dialga - 10.1.1.1	9558
onix	dialga - 10.1.1.1	9559
togepi	dialga - 10.1.1.1	9560
mew	dialga - 10.1.1.1	9561
Tor VM	dialga - 10.1.1.1	9562
geodude	dialga - 10.1.1.1	9563

[5] `http://amunhoney.sourceforge.net/`

Firewall rules are set in *arbok* and *dialga* to allow these flow exports, however, even if rules are still present for the tor monitoring VM, as it is not running anymore, no flows are exported.

These flows are still collected using nfsen, whose interface can be accessed at `http://dialga/nfsen/nfsen.php`.

3.4 leurrecom.org

The leurrecom.org project aims at getting a more realistic picture of the attacks happening on the Internet as well as their root causes (i.e. organized crime vs script kiddies) using unbiased quantitative data. For more than 3 years now, a worldwide distributed set of identical honeypots has been deployed in many different countries. All their tcpdump files are centralized in a database that all partners have access to for free. As of today, around 50 instances are running in almost 30 different countries covering the 5 continents.

Our contacts in the project are Marc Dacier and Corrado Leita. We followed the instruction that were provided by them to perform the installation with a custom Linux distribution based on a Fedora Core 7.

However, as our hardware was very recent, some devices were not well supported by the existing drivers. The SATA optical drive in the server was not recognized, and we had to use an external DVD drive to perform the installation, after trying to generate a USB stick, which failed because of hard-coded steps in the installation process.

Once the installation finalized, neither of the network cards (integrated Broadcom NetXtrem II 5716 or additional Intel Pro 1000 ET 82576) were supported. We had to install the appropriate driver for one of them. We decided to install manually the igb [6] driver (version 2.3.4) for the Intel Pro 1000 card. We installed via USB stick all the dependencies and configured the system to recognize the card as eth0 and eth1 instead of eth2 and eth3 by editing */etc/modprobe.conf* and */etc/udev/rules.d/60-net.rules*.

All the operation and maintenance tasks are performed on the centralized logging server of the project by the leurrecom.org team.

[6] `http://e1000.sf.net`

4 Experiments

4.1 Tor

We have been running a Tor relay router on *nidoran* for several months (migrated from *mankey*). Due to several abuse reports from CERT organizations of various countries (we were the output router in the anonimization process of attackers), we shut that VM down. It is still stored on the server, and can be started for one shot experiments.

During its run, we collected 195GB of tcpdump data (only the headers, not the data) and 5.8GB of network flows. We need now to study and analyze all these data.

It is possible to obtain a list of all Tor relays at a given timestamp via Tor's official data collection project [7]. It contains Tor relays descriptors archives since May 2004, and bridge descriptors archives, statistics, and information about the performances of the network.

Alongside to this data collection project, a tool called *Ernie* allows to generate graphs and save all types of archive data about the Tor network, such as relay descriptors into a postgresql 8.3 database. This could be very helpful when analyzing the collected data about the Tor network, or when correlating this information with the telescope itself.

4.2 Peer-to-peer monitoring

In the scope of the MAPE project, we have 2 VMs, *mape-manager* and *mape-manager-juan*, running on *geodude*. These VMs are used by Thibault Cholez for P2P monitoring and measurements in the KAD network. When performing large scale experiments, the SDSL bandwidth as well as the disk space on the VMs (20GB by default) were saturated. Therefore, we were using a dedicated computer plugged on the Free ADSL connection.

Another VM, *bittorrent-monitoring-01*, is running on *geodude* and is used by Juan Pablo Timpanaro for the same kind of experiments on the bittorrent network. We may face the same kind of problem than for KAD when performing large scale experiments.

For all these VMs, the postgresql servers running are accepting up to 350 simultaneous connections. To start the server, kernel parameters SHMMIN, SHMMAX and SHMALL must be set to the values specified in */etc/sysctl.conf*. To make sure that these VMs have sufficient memory and CPU, we boosted these parameters (4 VCPUs and 1GB of memory instead of A VCPU and 512MB of memory).

4.3 VoIP honeypots

In the scope of the VAMPIRE ANR Project, we have deployed several VoIP honeypots on the *nidoran* server used by Laurent Andrey:

voip-honeypot-01 Debian Linux

voip-honeypot-02 CentOS 4.8 for Orange SIP honeyd based honeypot

[7] `http://metrics.torproject.org/data.html`

voip-honeypot-03 Debian Linux for artemisa deployment

voip-honeypot-04 Debian Linux running Dionaea with SIP emulation module activated

All logs and outputs are stored on *dialga* at */data/users/vampire*. The vampire user on the server is in charge of getting the logs via rsync from the different honeypots.

4.4 SSH honeypot with University of Luxembourg

A custom SSH honeypot, hali, was deployed as a cooperation with the University of Luxembourg. It was an SSH tunnel between a Debian VM running on pikachu and the actual honeypot in Luxembourg. This honeypot is no longer running, but the VM is still stored on pikachu if required.

5 Future work

Even if the upgrade has been successfully performed, there are still some points that need some investigation.

First of all, we have collected and are still collecting lots of data, of many different kinds (pcap traces, network flows, attacks logs...), but we did not analyze them rigorously yet.

The telescope is running 4 different honeypots at the moment (nepenthes, dionaea, kippo and amun), but the Surfnetids has been designed to work with their own honeypots. Deploying a tunnelserver instance on *dialga* and Surfnet sensors on the server *togepi* would permit to finalize the upgrade and honeypots diversification.

Finally, in the experimentation environment, we still have to upgrade the VMWare Player on *mankey* to VMWare vSphere. If we want to open this environment to partners, Open Management Framework [8] must be investigated and maybe deployed if required. We may need as well to define and write an NDA or other kind of document.

[8] `http://omf.mytestbed.net/projects/omf/wiki`

6 Conclusion

We performed a full software upgrade of the platform and upgraded the main network connection to match the new requirements following the update. We have 81 sensors running based on 4 different honeypots, and still need to deploy 14 on the server *togepi*. We keep on collecting network traces and flow records together with the binaries and attacks details on the honeypots.

We have also developed the experimentation environment and performed various studies on various subjects (P2P monitoring and observation, VoIP honeypots...).

With the new honeypots, we obtained the following results the 11th November 2010:

Figure 4: Results for the 11th November 2010

Detected Connections	**Statistics**
Possible malicious attack	225 258
Malicious attack	53 337
Nepenthes	49
Amun	14 444
Dionaea	37 141
Kippo	1 703
Malware offered	52 235
Malware downloaded	11 858

Overall, since the 9th September 2008, the telescope suffered the following attacks:

Figure 5: Results since the 9th September 2008

Detected Connections	**Statistics**
Possible malicious attack	34 962 357
Malicious attack	2 296 567
Nepenthes	1 384 035
Amun	15 982
Dionaea	804 682
Kippo	91 868
Malware offered	2 222 058
Malware downloaded	3 352 845
Total number of unique binaries	125 605

We collected a total of 1 010GB of pcap dump traces and 20GB of network flow records.

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803