System Manual Project O

Prepared for

Version 0.1

Inhalt

Change control

Von Dreamlab Technologies AG
Datum 01. Oktober 2010
Betreff Iproxy Oman System Manual

to Projektleiter / Teilprojektleiter

Cc -

Dok ID	-
Dok Bezeichnung	System Manual Iproxy Oman
Version	0.1

Change control

Version	Datum	Ausführende Stelle	Bemerkungen/Art der Änderung
0.1	06.09.10	Richard Sademach Dreamlab Technologies AG	Initial Version

Prüfung

Version	Prüfdatum	Prüfende Stelle/n	Bemerkungen

1. Document Information

1.1 Purpose of the Document

The present document is the manual for the iProxy infrastructure, which describes the server systems, including their functions and comunication paths.

1.2 Delimitations

1.3 Scope of validity

The present document is only valid for the iProxy infrastructure that has been delivered as a result of project O. It is valid without time limitation until it is replaced by a new version.

1.4 Target audience

The present documents targets the system responsibles for the iProxy infrastructure.

1.5 Referenced Documents

-

1.6 Abbreviations

iProxy	Infection Proxy
NDP	Network Data Processor
RP	Radius Probe
ADMF	Administrative Mediation Fuction
ADMF-GUI	Administrative Mediation Function Graphical User Interface

2. iProxy System Overview

2.1 General

The Finfly iProxy infrastrucure consists of four different types of components:

- Two network data processors (NDP), parsing the data in bridged mode and initiating the data injection on demand,
- Two RADIUS probes (RP), parsing the RADIUS information and forwarding it to ADMF.
- The administrative mediation function (ADMF), controlling the NDPs and Rps
- The ADMF-GUI, allowing the operator/ administrator to control the infrastructure by a central user interface.

The infection process can be conducted in two different variants, both of them can either be initiated individually or in combination for one target:

1. **Binary infection:** The binary download mode is used to infect binaries that are downloaded from the internet by the configured target. In order to do this, the software analyzes the data streams on the NDPs at both of the internet exchanges (IX). As soon as a matching type of binary is downloaded, the infection mechanism is initiated, then it attaches loader and payload (trojan) to the binary.
2. **Update infection:** The update infection mode works by sending counterfeit server responses to predefined applications (for example iTunes, Winamp, OpenOffice and SimpleLite), when they are searching for updates. The NDPs analyze the target's data streams and parse the application's update request. Then, they send back a conterfeit response back to the application, informing it that the update wold be ready for download. The application subsequently starts the download of the update which will be treated with loader and payload (trojan) by the NDP.

All the iProxy components are connected by the jabber protocol with the jabber server running on the ADMF. All communication paths are encrypted for security reasons. Also all components of the iProxy infrastructure are based on daemontools, a collection of tools for the high availability management of UNIX services.

2.2 Overview of the infrastructure

2.2.1 Infrastructure schema

1x Bild gemäss Offerte mit Visio erstellen

2.2.2 Infrastruktur Kommunikationswege

1x Bild Visio Zusammensetzung unserer Komponenten mit Kommunikations wegen

2.2.3 Infrastruktur Management Netzwerk

1x Bild Visio Management Netzwerk

2.2.4 IP- und Hostinformationen

IP und Host Informationen			
Standort	**Host Name**	**IP Adresse**	**Beschreibung**
Tbd.	ADMF-GUI	Tbd.	
Tbd.	ADMF	Tbd.	
Tbd.	NDP01	Tbd.	
Tbd.	NDP02	Tbd.	
Tbd.	RP01	Tbd.	
Tbd.	RP02	Tbd.	

3. Server Installation

The following chapter describes setup of each server's hardware and the installed components

Dieses Kapitel beschreibt die Installationen und das Hardware Setup der einzelnen Server sowie deren installierten Komponenten.

3.1 ADMF-GUI (Adminstrative User Interface)

This machine hosts the graphical user interface for the management of the ADMF components. The user interface is the main tool for configuring infections, reviewing their status, etc.

3.1.1 Hardware

Server hardware configuration

Hardware	Description
Model	HP Compaq 8000 Elite Business PC
Processor	Intel® Core 2
Hard drive	1 x 250GB
Main Memory	2 x 1024MB
Network interfaces	2 x OnBoard RJ45
Mainboard	Intel® Q45 Express

3.1.2 Software

Software	Description
Operating system	Windows 7 Ultimate
Operating system kernel	Windows 7

3.1.3 Hard disk partitions

The system is equipped with hardware RAID and configured with the following partition scheme:

Partition	Purpose	Size	File system
C:	Root partition	250GB	NTFS

3.2 ADMF (central administrative function)

This server is the core component of the iProxy infrastructure. All parts of the infrastructure areme managed by the ADMF. Also, all infections are configured and initiated on the ADMF which is in constant communication with the other components of the iProxy infrastructure. It is constantly aware of the iProxy's status, and provides status information change messages (RADIUS / DHCP / successful infections, etc.) The information between the components is exchanged using the jabber protocol and encryption.

3.2.1 Hardware

Server hardware configuration:

Hardware	Description
Model	HP Proliant G6
Processors	2x Intel(R) Xeon(R) CPU X5550 @ 2.67GHz
Hard disk	3 x 146 GB SAS 2,5'' (Raid 5)
Main Memory	12GB
Network interfaces	4 x Broadcom NetXtreme II BCM5709
Management interface	1 x ILO
Mainboard	Intel 5520

3.2.2 Software

Software	Description
Operating system	Debian GNU/Linux 5.0 (Lenny)
Operating system kernel	2.6.26-2-amd64

3.2.3 Services

Service	Description
SSH	Secure remote shell (Remote administration)
Shorewall	High level firewall configuration tool (firewall)
syslog-ng	Syslog server (logging)
daemontools	Tools for the high availability management of the iProxy components
Ejabberd	Communication service for all iProxy infrastructure components
beam	Communication service for all iProxy infrastructure

	components
epmd	Communication service for all iProxy infrastructure components
Admf	Communication service for all iProxy infrastructure components

3.2.4 Hard disk partitions

The system is equipped with hardware RAID and configured with the following partition scheme:

Partition	Purpose	Size	File system
/dev/cciss/c0d0p1	/	80GB	ext3
/dev/cciss/c0d0p2	/home	150GB	ext3
/dev/cciss/c0d0p3	/var	57.6GB	ext3
/dev/cciss/c0d0p4	swap	6GB	swap

3.3 NDP01/02 (Iproxy component)

This server represents the network data processing component of the iProxy infrastructure. With the ADMF and ADMF GUI infections can be configured and activated on this component. As soon as the target is online and it's current IP address is either found by a RADIUS probe (RP01 and RP02) or already statically configured, the infection process is initiated. The two different variants for the infection are described above (see 2.1). Either one of them can be initiated seperately, or both simultaneously .

3.3.1 Hardware

Server hardware configuration:

Hardware	Description
Model	HP Proliant G7
Processors	1x Intel(R) Xeon(R) CPU X5650 @ 2.67GHz
Hard disks	3 x 146 GB SAS 2,5'' (Raid 5)
Main memory	12GB
Network interfaces	4 x Broadcom NetXtreme II BCM5709
Bypass Interface	1 x PE210G2BPi9-SR (Fiber / Multimode)
Management Interface	1 x ILO
Mainboard	Intel 5520

3.3.2 Software

Software	Description
Operating system	Debian GNU/Linux 5.0 (Lenny)
Operating system kernel	2.6.35-2-amd64

3.3.3 Services

Service	Description
SSH	Secure Remote Shell (Remote Admin)
Shorewall	High level firewall configuration tool (firewall)
syslog-ng	Syslog server (logging)
daemontools	Tools for the high availability management of the iProxy components
Ndp	Network data processing software

finfly_isp_proxy	Software for infecting binaries and updates

3.3.4 Hard disk partitions

The system is equipped with hardware RAID and configured with the following partition scheme:

Partition	Purpose	Size	File system
/dev/cciss/c0d0p1	/	80GB	ext3
/dev/cciss/c0d0p2	/home	150GB	ext3
/dev/cciss/c0d0p3	/var	57.6GB	ext3
/dev/cciss/c0d0p4	swap	6GB	swap

3.4 RP01/02 (iProxy component)

With this server the RADIUS authentications and the subsequent DHCP leases are tapped and recorded, to always be aware of the target's IP address. The iProxy software component sends these informations to the ADMF which alerts the NDPs. So it is excluded that the data streams of false targets is infected. This is only used if the IP address isn't assigned statically or configured in the ADMF GUI. The tapping is applied using a 1 GBit/s copper tap.

3.4.1 Hardware

Server hardware configuration:

Hardware	Description
Model	HP Proliant G6
Processors	2x Intel(R) Xeon(R) CPU X5550 @ 2.67GHz
Hard disks	3 x 146 GB SAS 2,5'' (Raid 5)
Main Memory	12GB
Network interfaces	4 x Broadcom NetXtreme II BCM5709
Management interface	1 x ILO
Mainboard	Intel 5520
Model	HP Proliant G6

3.4.2 Software

Software	Description
Operating system	Debian GNU/Linux 5.0 (Lenny)
Operating system kernel	2.6.35-2-amd64

3.4.3 Services

Service	Description
SSH	Secure Remote Shell (Remote Admin)
Shorewall	High-Level firewall configuration tool (Firewall)
syslog-ng	Syslog server (logging)
daemontools	Tools zur Verwaltung von Unix-Services
Ndp	Software zum network data prcessing
finfly_isp_proxy	Software zur Infektion der Binaries / Updates

3.4.4 Hard disk partitions

The system is equipped with hardware RAID and configured with the following partition scheme:

Partition	Purpose	Size	File system
/dev/cciss/c0d0p1	/	80GB	ext3
/dev/cciss/c0d0p2	/home	150GB	ext3
/dev/cciss/c0d0p3	/var	57.6GB	ext3
/dev/cciss/c0d0p4	swap	6GB	swap

4. System configuration

The following chapter describes the configuration of the systems and the running services.

4.1 ADMF-GUI

4.1.1 Network interfaces

Interface	IP-address	Network
Local Area Connection 1	Tbd.	Management network

4.1.2 Routing

Destination	Gateway	Mask	Interface
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.

4.1.3 FinFly GUI

Tbd.

4.2 ADMF

4.2.1 Network interfaces

Interface	IP-Address	Network
eth0	Tbd.	Management network
ilo01	Tbd.	Management network

4.2.2 Routing

Destination	Gateway	Mask	Interface
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.

4.2.3 Configuration of the hosts file

In order to ensure the correct behaviour of the iProxy components, on every server hosting a component, the IP address of the cetral ADMF server must be registerd. This is done in the following configuration file:

/etc/hosts

```
127.0.0.1       localhost
192.168.123.155 admf


# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

4.2.4 Network configuration

The network interfaces are configured with the following configuration file:

/etc/network/interfaces

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
        address 192.168.123.155
        netmask 255.255.255.0
        network 192.168.123.0
        broadcast 192.168.123.255
        gateway 192.168.123.1
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 208.67.222.222
        up route add -net 192.168.41.0/24 gw 192.168.123.1
        up route add -net 192.168.124.0/24 gw 192.168.123.1
```

4.2.5 ADMF

The configuration of the iProxy ADMF components resides in the following configuration file. It is used to configure the target database, the ADMF's jabber ID and password, the ADMF GUI's jabber ID, the NDPs' jabber IDs and the Radius probes' Jabber IDs. Further, it defines the necessaary environment and path variables for the ADMF component.

/home/iproxy/service/admf/etc/instance.conf

```
# -*- coding: utf-8 -*-

export VERBOSE=0
```

```
# ADMF
# the INSTANCE_DIR variable is set by the daemontools launch script
export DATA_DIR_PATH="${INSTANCE_DIR}/data"
export DB_FILE_NAME="admf.db"

# ADMF manager
export ADMF_JID="admf@admf"
export ADMF_SECRET="XXXXXXXXX"

# ADMF<->NDP
export NDP_JIDs="ndp01@admf ndp02@admf"

# ADMF<-GUI
export GUI_JID="gui@admf"

# ADMF<->RPROBEs
export RP_JIDs="rp01@admf rp02@admf"



# settings below this line are autogenerated by the provision script
# and should need no change unless you know what you are doing
export PYTHONPATH="/home/iproxy/code:/home/iproxy/code/lib/python"
export EXEC_PATH="/home/iproxy/code/finfly/admf.py"
#export INSTANCE_NAME="admf"
```

4.2.6 Ejabberd configuration

The ejabberd is configured with the following configuration file:

/etc/ejabberd/ejabberd.cfg

```
%%%
%%%     Debian ejabberd configuration file
%%%     This config must be in UTF-8 encoding
%%%
%%% The parameters used in this configuration file are explained in more detail
```

```
%%% in the ejabberd Installation and Operation Guide.
%%% Please consult the Guide in case of doubts, it is available at
%%% /usr/share/doc/ejabberd/guide.html

%%%   ==================================
%%%   OVERRIDE OPTIONS STORED IN DATABASE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Options which are set by Debconf and managed by ucf

%% Admin user
{acl, admin, {user, "admin", "admf"}}.

%% Hostname
{hosts, ["admf"]}.

%%%   =========
%%%   DEBUGGING

%%
%% loglevel: Verbosity of log files generated by ejabberd.
%% 0: No ejabberd log at all (not recommended)
%% 1: Critical
%% 2: Error
%% 3: Warning
%% 4: Info
%% 5: Debug
%%
{loglevel, 4}.

%%%   ===============
%%%   LISTENING PORTS

%%
%% listen: Which ports will ejabberd listen, which service handles it
%% and what options to start it with.
%%
{listen,
```

```
 [
  {5222, ejabberd_c2s, [
                        {access, c2s},
                        {shaper, c2s_shaper},
                        {max_stanza_size, 65536},
                        starttls, {certfile, "/etc/ejabberd/ejabberd.pem"}
                       ]},
  {5269, ejabberd_s2s_in, [
                           {shaper, s2s_shaper},
                           {max_stanza_size, 131072}
                          ]},
  {5280, ejabberd_http, [
                         http_poll,
                         web_admin
                        ]}

 ]}.

%%
%% s2s_use_starttls: Enable STARTTLS + Dialback for S2S connections.
%% Allowed values are: true or false.
%% You must specify a certificate file.
%%
{s2s_use_starttls, true}.

%%
%% s2s_certfile: Specify a certificate file.
%%
{s2s_certfile, "/etc/ejabberd/ejabberd.pem"}.

%%%   ==============
%%%   AUTHENTICATION

%%
%% auth_method: Method used to authenticate the users.
%% The default method is the internal.
%% If you want to use a different method,
```

```
%% comment this line and enable the correct ones.
%%
{auth_method, internal}.

%%%   ===============
%%%   TRAFFIC SHAPERS

%%
%% The "normal" shaper limits traffic speed to 1.000 B/s
%%
{shaper, normal, {maxrate, 1000}}.

%%
%% The "fast" shaper limits traffic speed to 50.000 B/s
%%
{shaper, fast, {maxrate, 50000}}.

%%%   ====================
%%%   ACCESS CONTROL LISTS
%%
%% Local users: don't modify this line.
%%
{acl, local, {user_regexp, ""}}.

%%%   ============
%%%   ACCESS RULES

%% Define the maximum number of time a single user is allowed to connect:
{access, max_user_sessions, [{10, all}]}.

%% This rule allows access only for local users:
{access, local, [{allow, local}]}.

%% Only non-blocked users can use c2s connections:
{access, c2s, [{deny, blocked},
               {allow, all}]}.
```

```
%% For all users except admins used "normal" shaper
{access, c2s_shaper, [{none, all}]}.

%% For all S2S connections used "fast" shaper
{access, s2s_shaper, [{fast, all}]}.

%% Only admins can send announcement messages:
{access, announce, [{allow, admin}]}.

%% Only admins can use configuration interface:
{access, configure, [{allow, admin}]}.

%% Admins of this server are also admins of MUC service:
{access, muc_admin, [{allow, admin}]}.

%% All users are allowed to use MUC service:
{access, muc, [{allow, all}]}.

%% No username can be registered via in-band registration:
%% To enable in-band registration, replace 'deny' with 'allow'
% (note that if you remove mod_register from modules list then users will not
% be able to change their password as well as register).
% This setting is default because it's more safe.
{access, register, [{deny, all}]}.

%% Everybody can create pubsub nodes
{access, pubsub_createnode, [{allow, all}]}.

%%%   DEFAULT LANGUAGE

%%
%% language: Default language used for server messages.
%%
{language, "en"}.



%%%   =======
```

```
%%%   MODULES


%%
%% Modules enabled in all ejabberd virtual hosts.
%%
{modules,
 [
  {mod_adhoc,    []},
  {mod_announce, [{access, announce}]}, % requires mod_adhoc
  {mod_caps,     []},
  {mod_configure,[]}, % requires mod_adhoc
  {mod_ctlextra, []},
  {mod_disco,    []},
  %%{mod_echo,   [{host, "echo.localhost"}]},
  {mod_irc,      []},
  {mod_last,     []},
  {mod_muc,      [
                  %%{host, "conference.@HOST@"},
                  {access, muc},
                  {access_create, muc},
                  {access_persistent, muc},
                  {access_admin, muc_admin},
                  {max_users, 500}
                 ]},
  %%{mod_muc_log,[]},
  {mod_offline,  []},
  {mod_privacy,  []},
  {mod_private,  []},
  {mod_proxy65,  [
                  {access, local},
                  {shaper, c2s_shaper}
                 ]},
  {mod_pubsub,   [ % requires mod_caps
                  {access_createnode, pubsub_createnode},
                  {plugins, ["default", "pep"]}
                 ]},
  {mod_register, [
```

```
                %%
                %% After successful registration, the user receives
                %% a message with this subject and body.
                %%
                {welcome_message, {"Welcome!",
                                   "Welcome to a Jabber service powered by Debian. "
                                   "For information about Jabber visit "
                                   "http://www.jabber.org"}},
                %% Replace it with 'none' if you don't want to send such message:
                %%{welcome_message, none},

                %%
                %% When a user registers, send a notification to
                %% these Jabber accounts.
                %%
                %%{registration_watchers, ["admin1@example.org"]},

                {access, register}
               ]},
 {mod_roster,   []},
 %%{mod_service_log,[]},
 {mod_shared_roster,[]},
 {mod_stats,    []},
 {mod_time,     []},
 {mod_vcard,    []},
 {mod_version,  []}
]}.
```

4.2.7 Firewall configuration

For the configuration of the firewall rules the shorewall software is used, please refer to Chapter 5.1 for further descriptions.

Interfaces

Zone	Interface	IP Address	Parameters
net	eth0	detect	tcpflags,logmartians,nosmurfs

Policies

Source	Destination	Policy	Log Level
$FW	Net	ACCEPT	
Net	$FW	DROP	Info
Net	All	DROP	Info
Net	All	DROP	Info
All	All	REJECT	Info

Zones

Zone	Type
fw	firewall
net	ipv4

Rules

Aktion	Source zone	Target zone	Protocol	Target port	Source port
Ping/DROP	Net	$FW			
ACCEPT	Net	$FW	TCP	62200	
ACCEPT	NET	$FW	TCP	5222	
ACCEPT	$FW	Net	ICMP		

4.3 NDP01

4.3.1 Network interfaces

Interface	IP-Address	Network
eth0	Tbd.	Management Network
br0	Tbd.	Bridged Network
ilo01	Tbd.	Management Network

4.3.2 Routing

Destination	Gateway	Mask	Interface
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.

4.3.3 Configuration of the hosts file

In order to ensure the correct behaviour of the iProxy components, on every server hosting a component, the IP address of the cetral ADMF server must be registerd. This is done in the following configuration file:

/etc/hosts

```
127.0.0.1       localhost
192.168.123.155 admf

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

4.3.4 Network configuration

The network interfaces are configured with the following configuration file:

/etc/network/interfaces

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.123.153
        netmask 255.255.255.0
        network 192.168.123.0
        broadcast 192.168.123.255
        up route add -net 192.168.41.0/24 gw 192.168.123.1
        up route add -net 192.168.124.0/24 gw 192.168.123.1

auto br0
iface br0 inet static
        bridge_ports eth4 eth5
        address 10.0.0.200
        netmask 255.255.255.0
        network 10.0.0.0
        broadcast 10.0.0.255
        gateway 10.0.0.1
        up /root/set_irq_affinity.sh eth4
        up /root/set_irq_affinity.sh eth5
        up /root/set_watchdog_timer.sh eth4
```

4.3.5 Configuration of the NDP component

The iProxy NDP component is configured with the following configuration file.

In this configuration file the verbosity level, the data directory for the infection component and the NDP interfaces are defined. Optionally, two bridges could be

configured, but this is no out-of-the-box feature of the present version. In order to do so, the jabber ID and it's password, the environment variables for the infection components, the ADMF communication and the NDP componets must be edited (see bottom of file).

/home/iproxy/service/ndp01/etc/instance.conf

```
# -*- coding: utf-8 -*-

export VERBOSE=0

export SERVICE_DIR_PATH="/etc/service"
# the INSTANCE_DIR variable is set by the daemontools launch script
export DATA_DIR_PATH="${INSTANCE_DIR}/data"
export UPDATES_DIR_NAME="application-upgrade"

# NDP
export TPROXY_PORT=3129
export IPTABLES_PATH="/home/iproxy/code/sbin/iptables"
export TGT_IF="eth4"
export INET_IF="eth5"
export TGT_IF1="eth4"
export INET_IF1="eth5"
export TGT_IF2="eth8"
export INET_IF2="eth9"

# NDP manager
export NDP_JID="ndp01@admf"
export NDP_SECRET="XXXXXXXXXX"

# NDP<->IPROXY
export IPROXY_DIR_PATH="/home/chrootusers/home/gamma/finfly_isp_proxy"
export IPROXY_USER="gamma"
export NDP_IP="127.0.0.1"
export NDP_INF_PORT=30001
export INF_IP="127.0.0.1"
```

```
export INF_NDP1_PORT=30002
export INF_NDP2_PORT=30003

# NDP<->ADMF
export ADMF_JID="admf@admf"



# settings below this line are autogenerated by the provision script
# and should need no change unless you know what you are doing
export PYTHONPATH="/home/iproxy/code:/home/iproxy/code/lib/python"
export EXEC_PATH="/home/iproxy/code/finfly/ndp.py"
#export INSTANCE_NAME="ndp01"
```

4.3.6 Configuration of the iProxy component

The iProxy component is configured with the following configuration file.

It defines the environment variables of the communication between the NDP and the infection component. The environment variables for the iProxy component's behavoiur is also defined in this configuration file.

/home/iproxy/service/iproxy/etc/instance.conf

```
# -*- coding: utf-8 -*-

# IProxy<->NDP
export NDP_IP="127.0.0.1"
export NDP_INF_PORT=30001
export INF_NDP1_PORT=30002
export INF_NDP2_PORT=30003



# settings below this line are autogenerated by the provision script
# and should need no change unless you know what you are doing
export PYTHONPATH="/home/iproxy/code:/home/iproxy/code/lib/python"
export EXEC_PATH="/home/iproxy/code/finfly/iproxy.py"
```

```
#export INSTANCE_NAME="iproxy"
```

4.3.7 Finfly ISP Proxy Komponente Konfiguration

The Finfly ISP Proxy component is configured with the following configuration file.

Because the Finfly ISP Proxy component runs in a separate chroot environment, the daemon tools can only ensure and control starting and stopping of the application.

Gamma muss Rest dokumentieren ...

/home/iproxy/service/finfly_isp_proxy/etc/instance.conf

```
export EXEC_PATH="/usr/sbin/jk_chrootlaunch -u gamma -g gamma -j
/home/chrootusers --exec /bin/bash -- /launch_finfly_isp_proxy"
```

4.3.8 Chroot jail

For running the infection component in a secure environment, the software jailkit (http://olivier.sessink.nl/jailkit/)is used, together with rbash. The secure environment is as well linked to the daemon tools to provide a very high level of availability.

4.3.9 Firewall configuration

The configuration of the NDP's firewall is integrated directly into the iProxy component, because the firewall rules are adopted dynamically to the targets configured in the ADMF GUI (iptables and etables simultaneously).

4.4 NDP02

4.4.1 Netzwerkinterfaces

Interface	IP-Address	Network
eth0	Tbd.	Management Network
br0	Tbd.	Bridged Network
ilo01	Tbd.	Management Network

4.4.2 Routing

Destination	Gateway	Mask	Interface
Tbd.	Tbd.	Tbd.	Tbd.

Destination	Gateway	Mask	Interface
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.

4.4.3 Configuration of the hosts file

In order to ensure the correct behaviour of the iProxy components, on every server hosting a component, the IP address of the central ADMF server must be registerd. This is done in the following configuration file:

/etc/hosts

```
127.0.0.1       localhost
192.168.123.155 admf

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

4.4.4 Network configuration

The network interfaces are configured with the following configuration file:

/etc/network/interfaces

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
```

```
iface eth0 inet static
        address 192.168.123.154
        netmask 255.255.255.0
        network 192.168.123.0
        broadcast 192.168.123.255
        up route add -net 192.168.41.0/24 gw 192.168.123.1
        up route add -net 192.168.124.0/24 gw 192.168.123.1


auto br0
iface br0 inet static
        bridge_ports eth4 eth5
        #address 10.42.42.200
        address 10.0.0.200
        netmask 255.255.255.0
        #network 10.42.42.0
        network 10.0.0.0
        #broadcast 10.42.42.255
        broadcast 10.0.0.255
        #gateway 10.42.42.1
        gateway 10.0.0.1
        up /root/set_irq_affinity.sh eth4
        up /root/set_irq_affinity.sh eth5
        up /root/set_watchdog_timer.sh eth4
```

4.4.5 Configuraiton of the NDP component

The iProxy NDP component is configured with the following configuration file.

In this configuration file the verbosity level, the data directory for the infection component and the NDP interfaces are defined. Optionally, two bridges could be configured, but this is no out-of-the-box feature of the present version. In order to do so, the jabber ID and it's password, the environment variables for the infection components, the ADMF communication and the NDP componets must be edited (see bottom of file).

/home/iproxy/service/ndp02/etc/instance.conf

```
# -*- coding: utf-8 -*-
```

```
export VERBOSE=0

export SERVICE_DIR_PATH="/etc/service"
# the INSTANCE_DIR variable is set by the daemontools launch script
export DATA_DIR_PATH="${INSTANCE_DIR}/data"
export UPDATES_DIR_NAME="application-upgrade"

# NDP
export TPROXY_PORT=3129
export IPTABLES_PATH="/home/iproxy/code/sbin/iptables"
export TGT_IF="eth4"
export INET_IF="eth5"
export TGT_IF1="eth4"
export INET_IF1="eth5"
export TGT_IF2="eth8"
export INET_IF2="eth9"

# NDP manager
export NDP_JID="ndp02@admf"
export NDP_SECRET="XXXXXXXXXX"

# NDP<->IPROXY
export IPROXY_DIR_PATH="/home/chrootusers/home/gamma/finfly_isp_proxy"
export IPROXY_USER="gamma"
export NDP_IP="127.0.0.1"
export NDP_INF_PORT=30001
export INF_IP="127.0.0.1"
export INF_NDP1_PORT=30002
export INF_NDP2_PORT=30003

# NDP<->ADMF
export ADMF_JID="admf@admf"
```

```
# settings below this line are autogenerated by the provision script
# and should need no change unless you know what you are doing
export PYTHONPATH="/home/iproxy/code:/home/iproxy/code/lib/python"
export EXEC_PATH="/home/iproxy/code/finfly/ndp.py"
#export INSTANCE_NAME="ndp02"
```

4.4.6 Configuration of the iProxy component

The iProxy component is configured with the following configuration file. It defines the environment variables of the communication between the NDP and the infection component. The environment variables for the iProxy component's behavior is also defined in this configuration file.

/home/iproxy/service/iproxy/etc/instance.conf

```
# -*- coding: utf-8 -*-


# IProxy<->NDP
export NDP_IP="127.0.0.1"
export NDP_INF_PORT=30001
export INF_NDP1_PORT=30002
export INF_NDP2_PORT=30003



# settings below this line are autogenerated by the provision script
# and should need no change unless you know what you are doing
export PYTHONPATH="/home/iproxy/code:/home/iproxy/code/lib/python"
export EXEC_PATH="/home/iproxy/code/finfly/iproxy.py"
#export INSTANCE_NAME="iproxy"
```

4.4.7 Configuration of the Finfly ISP Proxy component

The Finfly ISP Proxy component is configured with the following configuration file. Because the Finfly ISP Proxy component runs in a separate chroot environment, the daemon tools can only ensure and control starting and stopping of the application.

Gamma muss Rest dokumentieren ...

/home/iproxy/service/finfly_isp_proxy/etc/instance.conf

```
export EXEC_PATH="/usr/sbin/jk_chrootlaunch -u gamma -g gamma -j
/home/chrootusers --exec /bin/bash -- /launch_finfly_isp_proxy"
```

4.4.8 Chroot jail

For running the infection component in a secure environment, the software jailkit (http://olivier.sessink.nl/jailkit/)is used, together with rbash. The secure environment is as well linked to the daemon tools to provide a very high level of availability.

4.4.9 Firewall configuration

The configuration of the NDP's firewall is integrated directly into the iProxy component, because the firewall rules are adopted dynamically to the targets configured in the ADMF GUI (iptables and etables simultaneously).

4.5 RP01

4.5.1 Network interfaces

Interface	IP-Address	Network
eth0	Tbd.	Management Network
bond0	1.2.3.4	Tapping Interface für Radius Kommunikation
ilo01	Tbd.	Management Network

4.5.2 Routing

Destination	Gateway	Mask	Interface
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.

4.5.3 Configuration of the hosts file

In order to ensure the correct behaviour of the iProxy components, on every server hosting a component, the IP address of the central ADMF server must be registerd. This is done in the following configuration file

/etc/hosts

```
127.0.0.1       localhost
192.168.123.155 admf


# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

4.5.4 Network configuration

The network interfaces are configured with the following configuration file:

/etc/network/interfaces

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
        address 192.168.123.151
        netmask 255.255.255.0
        network 192.168.123.0
        broadcast 192.168.123.255
        gateway 192.168.123.1
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 208.67.222.222



auto bond0
iface bond0 inet static
    address 1.2.3.4
    netmask 255.255.255.0
    network 1.2.3.0
    gateway 1.2.3.1
    slaves eth4 eth5
    bond_mode broadcast
    bond_miimon 100
    bond_downdelay 200
    bond_updelay 200
```

4.5.5 RP component configuration

The iProxy RP component is configured with the following configuration file. It configures the verbosity level, the tap/bond interface which is capturing the data and the port for

sending and receiving RADIUS data. Further, it defines the RADIUS probe jabber ID and password used for ADMF authentication, as well as the environment variables used by the radius probe component.

/home/iproxy/service/ndp01/etc/instance.conf

```
# -*- coding: utf-8 -*-


export VERBOSE=0


# RADIUS probe
export RADIUS_IF="bond0"
export RADIUS_PORT=1813


# RADIUS probe manager
export RP_JID="rp01@admf"
export RP_SECRET="XXXXXXXXX"


# RADIUS<->ADMF
export ADMF_JID="admf@admf"




# settings below this line are autogenerated by the provision script
# and should need no change unless you know what you are doing
export PYTHONPATH="/home/iproxy/code:/home/iproxy/code/lib/python"
export EXEC_PATH="/home/iproxy/code/finfly/radius.py"
#export INSTANCE_NAME="rp01"
```

4.5.6 Firewall configuration

For the configuration of the firewall rules the shorewall software is used, please refer to chapter 5.1 for further descriptions.

Interfaces

Zone	Interface	IP Adresse	Parameters

net	eth0	detect	tcpflags,logmartians,nosmurfs

Policies

Source	Destination	Policy	Log Level
$FW	Net	ACCEPT	
Net	$FW	DROP	Info
Net	All	DROP	Info
Net	All	DROP	Info
All	All	REJECT	Info

Zones

Zone	Type
fw	firewall
net	ipv4

Rules

Aktion	Source zone	Target zone	Protocol	Target port	Source port
Ping/DROP	Net	$FW			
ACCEPT	Net	$FW	TCP	62200	
ACCEPT	$FW	Net	ICMP		

4.6 RP02

4.6.1 Network interfaces

Interface	IP-Address	Network
eth0	Tbd.	Management Network
bond0	1.2.3.4	Tapping Interface für Radius Kommunikation
ilo01	Tbd.	Management Network

4.6.2 Routing

Destination	Gateway	Mask	Interface
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.
Tbd.	Tbd.	Tbd.	Tbd.

4.6.3 Configuration of the hosts file

In order to ensure the correct behaviour of the iProxy components, on every server hosting a component, the IP address of the central ADMF server must be registerd. This is done in the following configuration file.

/etc/hosts

```
127.0.0.1       localhost
192.168.123.155 admf


# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

4.6.4 Network configuration

The network interfaces are configured with the following configuration file:

/etc/network/interfaces

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
        address 192.168.123.152
        netmask 255.255.255.0
        network 192.168.123.0
        broadcast 192.168.123.255
        gateway 192.168.123.1
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 208.67.222.222



auto bond0
iface bond0 inet static
    address 1.2.3.4
    netmask 255.255.255.0
    network 1.2.3.0
    gateway 1.2.3.1
    slaves eth4 eth5
    bond_mode broadcast
    bond_miimon 100
    bond_downdelay 200
    bond_updelay 200
```

4.6.5 RP component configuration

The iProxy RP component is configured with the following configuration file. It configures the verbosity level, the tap/bond interface which is capturing the data and the port for

sending and receiving RADIUS data. Further, it defines the RADIUS probe jabber ID and password used for ADMF authentication, as well as the environment variables used by the radius probe component.

/home/iproxy/service/rp02/etc/instance.conf

```
# -*- coding: utf-8 -*-


export VERBOSE=0


# RADIUS probe
export RADIUS_IF="bond0"
export RADIUS_PORT=1813


# RADIUS probe manager
export RP_JID="rp02@admf"
export RP_SECRET="XXXXXXXXX"


# RADIUS<->ADMF
export ADMF_JID="admf@admf"




# settings below this line are autogenerated by the provision script
# and should need no change unless you know what you are doing
export PYTHONPATH="/home/iproxy/code:/home/iproxy/code/lib/python"
export EXEC_PATH="/home/iproxy/code/finfly/radius.py"
#export INSTANCE_NAME="rp02"
```

4.6.6 Firewall configuration

For the configuration of the firewall rules the shorewall software is used, please refer to chapter 5.1 for further descriptions.

Interfaces

Zone	Interface	IP-Address	Parameters

net	eth0	detect	tcpflags,logmartians,nosmurfs

Policies

Source	Destination	Policy	Log Level
$FW	Net	ACCEPT	
Net	$FW	DROP	Info
Net	All	DROP	Info
Net	All	DROP	Info
All	All	REJECT	Info

Zones

Zone	Type
fw	firewall
net	ipv4

Rules

Aktion	Source zone	Target zone	Protocol	Target port	Source port
Ping/DROP	Net	$FW			
ACCEPT	Net	$FW	TCP	62200	
ACCEPT	$FW	Net	ICMP		

5. Configuration advices

5.1 Shorewall

Shorwall is a useful tool fir configuring the firewall built-in in Linux (iptables/netfilter). Instead of confuguring single, cryptic firewall rules, Shorewall works with zones, default policies and rules controlling the traffic between the zones.

5.1.1 Configuration files

Shorewall saves it's configurations in /etc/shorewall. For the iProxy environment, the following files are relevant:

- `shorewall.conf` – Basic settings for Shorewall
- `zones` – Definition of zone names
- `interfaces` – Mapping between zones and network interfaces
- `policy` – Default policies for traffic between the zones
- `rules` – rules for specified and customized traffic

As an example, a simple configuration is shown below. First the zone names are defined:

```
## /etc/shorewall/zones
fw          firewall
inet        ipv4
lan         ipv4
```

Thus the zone “fw” is defined. The paramenter “firewall” specifies that the present system is the system running Shorewall. Further, the zones “inet” and “lan”, used for IPv4 traffic, are defined.

The next step assigns the interfaces to the zones:

```
## /etc/shorewall/interfaces
inet eth0   detect
lan   eth1  detect
```

This defines that eth0 belongs to the “inet” zone and eth1 to the “lan” zone.

Now the default policies can be defined:

```
## /etc/shorewall/policy

# Firewall may connect to the zones inet and lan without limitations
fw    inet  ACCEPT
fw    lan   ACCEPT
```

```
# Traffic from inet or lan to fw wis dropped. Further, dropped traffic is logged,
using the log level “info”.
inet  fw    DROP        info
lan   fw    DROP        info

# Disallow any communication between lan and inet.
inet  lan   DROP        info
lan   inet  DROP        info

# Last rule, dropping all traffic from all zones to all zones.
all   all   DROP        info
```

Wth the polices described hereby, alread much is achieved: The firewall can connect to both zones, while the zones “inet” and “lan” aren't allowed to connect to the firewall. In order to allow acces to defined services, correspondig rules are created:

```
## /etc/shorewall/rules

# Hosts in the zone “lan” are allowed to connect to the firewall using SSH and
SMTP (Ports 22 and 25)

ACCEPT      lan   fw    tcp   22
ACCEPT      lan   fw    tcp   25

# The same for DNS (udp port 53)
ACCEPT      lan   fw    udp   53

# Hosts in the zone “lan” are allowed to ping the firewall (ICMP-Type 8)
ACCEPT      lan   fw    icmp  8

# Hosts in the zone inet are allowed to connect to the firewall using HTTPS.

ACCEPT      inet  fw    tcp   443

# Hosts in the zone “lan” are allowed to connect to any HTTP and HTTPS service
in the zone “inet”
ACCEPT      lan   inet  tcp   80,443
```

Shorewall translates these simple configuration options into rules for netfilter or iptables. So the biggest part of the firewall's complexity isn't exposed to the system administrator, allowing him to concetrate specifically on the logics of creating rule sets.

5.2 E-mail distribution

On each system. Logwatch and logcheck are installed. These tools examine the system log files and send exceptional log messages by mail. For this reason, an E-mail distribution can be configured to spread these system notification mails.

Bsically, all systems can be configured with nulmailer. The following configuration would be possible to send mails from all systems to the address example@example.org.

Definition of the Mail server that will distribute the messages:

/etc/nullmailer/remotes

```
mailer.example.org
```

Name of the source system for mail delivery:

/etc/nullmailer/me

```
admf.example.org
```

The system's sender address:

/etc/nullmailer/adminaddr

```
syscheck@example.org
```

5.3 SSH

SSH is used for system administration. The following requirements to it's configuraiton are defined and implemented by the iProxy environment:

- Change default port 22 to 62200 (this prevents the system rom being found by automated SSH scanning tools)
- Only allow SSH version 2
- Direct root access is deactivated, a login with normal user accounts is fored.

6. Miscellaneous HOWTOs

6.1 Restart shorewall

In order to start the firewall (Shorewall), the folliwing command is used:

Um die Firewall (Shorewall) zu starten wird folgendes Kommando ausgeführt:

```
root@host:~# shorewall safe-restart
```

Thus, the firewall is started into a safe mode and applies the defined rules inly in case they function witout problems and the user allowes the application of the new ruleset by pressing “y”. If the firewall shouldn' start up, or an error in the system configuration is found, the previous firewall rule state is restored.

Also, the startup is aborted and the previous rules are restored, if the user doesn't confirm the new riles by pressing “y” within 60 seconds. This allows the firewall to be configured by remote login without the risk of being locked out.

6.2 Viewing log files

In the direcory /var/log/ several log files can be read and analyzed using the following commands:

```
root@host:~# tail -n 500 /var/log/messages
```

Thus, the last 500 lines of the «messages» log file are displayed.

```
root@host:~# tail -f /var/log/messages
```

This displays the most recent log output of the «messages» in real time.

6.3 Restart of the iProxy service

The components of the iProxy services reside in the following directory on every server:

/home/iproxy/service/

Within this direcory, a further directory is created for each service. In there, the configurations and data reside in a structure defined by deamon tools.

In order to start the corresponding service, the directory of each software component must be linked to /etc/service. This can be done by the following command:

```
cd /home/iproxy/service

ln -s rp01 /etc/service
```

Then, the service is started and will be restarted automatically with every machine reboot or possible crash of any iProxy component.

To stop the service permanently, the symlink must be removed from the etc/service directoy.

To only stop the service temporarily, or to restart it, the following commands are used:

To start the service:

```
svc -t /etc/service/rp01
```

To stop the service:

```
svc -d /etc/service/rp01
```

To restart the service:

```
svc -u /etc/service/rp01
```

7. Co-Betrieb

Not yet defined

8. Maintenance and suport

8.1 Software maintenance

To assure the actuality and security of the iProxy environment, it's undergoing a maintenance at defined time windows, when the most recent software versions will be installed. Of course, the software is under constant development and errors are recovered and new features added. Nevertheless, maintenance and co-operation if the infrastructure must be defined with the client individually.

8.2 Contact address of Dreamlab Technologies AG

Dreamlab Technologies AG
Monbijoustrasse 36
3011 Bern

Tel. +4131 / 398 66 66
Fax. +4131 / 398 66 69

For general questions: contact@dreamlab.net

For technical questions: Fragen: richard.sademach@dreamlab.net