Wago Firewall Analyse

Es wird eine SW Bestands- und Funktions- Analyse der bestehenden Wago Firewall vorgenommen.

Typ der Firewall: **Personal Firewall** oder **extern Firewall**

Beteiligte NetzwerkKomponenten

- TCP/IP: IP-Adress + Port (MAC Adresse)
- UDP/IP: IP-Adress + Port (MAC Adresse)
- ICMP
- FTP
- HTTP
- NAT
- Proxy

SW Bestand

Folgende SW Komponenten bilden die Wago Firewall

SW Funktion

Nachfolgend wird die Funktion der SW Komponenten und ihre Zusammenarbeit beschrieben

Angewendete Techniken

Welche Techniken werden angewendet

- Stateful Packet Inspection
- SSL Deep Packet Inspection
- Man in the Middle
-

Transparenz

- Eine Seite Transparent
- Beide Seiten transparent
- Unsichtbar

Regelwerk

Die Methode basiert auf Mandatory Access Control. Die Reihenfolge der Regeln ist relevant, weil die erste passende Regel genutzt wird.

- Drop (lokal verworfen)
- Reject(abgelehnt z.B. über ICMP)
- ACCEPT, ALLOW, PASS oder FORWARD (erlaubt)
-

xtables-legacy-multi

/usr/sbin/xtables-legacy-multi

Benutzt das getsockopt/setsockopt-based kernel interface. Es sollte nur über die Subcommands genutzt werden.

Valid subcommands:

- iptables
- main4
- iptables-save
- save4
- iptables-restore
- restore4
- iptables-legacy
- iptables-legacy-save
- iptables-legacy-restore
- iptables-xml
- xml

ebtables

Definiert eine Ethernet bridge whitelist. Es werden Ethernet Frames untersucht. Basis aller EB Regeln ist

ebwliste.rls

Diese Regeln können mit einem Editor angepasst werden, ipbase.rls wir von keinem Tool generiert. Die Regeln sind statisch und nicht während der Laufzeit änderbar. Es können jedoch mittels

iptables-restore -n < {aux_rules}.rls

Mittels {schema}.xsl und {doc}.xml können neue EB Regeln erzeugt werden. Siehe Services. (z.B. ebwlist.xml und ebwlist.xsl)

Alle EB Regeln werden werden von folgenden Script behandelt :

sh /etc/firewall/ebtables/ebfirewall.sh [--disable]

Die EB Regeln **ebwlist.aa.rls** wird bei disable ausgeführt.

Ein Interface zur Modifikation der XML Docs wird in ***ebtables::process()*** im Modul process_ebtables bereitgestellt. Siehe hierzu die Kommandozeilenparameter ***firewall config-tool*** der firewall Applikation.

Es ist das Tool ebtables installiert: /usr/sbin/ebtables

iptables

Mit Hilfe von iptables wird Netfilter, der IP-Paketfilter des Linux-Kernels konfiguriert

Es ist das Tool ebtables installiert: /usr/sbin/iptables -> xtables-legacy-multi

Basis aller IP Regeln ist

ipbase.rls

Diese Regeln können mit einem Editor angepasst werden, ipbase.rls wir von keinem Tool generiert. Die Regeln sind statisch und nicht während der Laufzeit änderbar. Es können jedoch mittels

iptables-restore -n < {aux_rules}.rls

Mittels {schema}.xsl und {doc}.xml können neue IP Regeln erzeugt werden. Siehe Services. z.B. ipcmn.xml / ip-cmn.xsl)

Alle IP Regeln (in iptables und services) werden von folgenden Script behandelt :

sh /etc/firewall/iptables/ipfirewall.sh [--apply|--disable|--service]

Ein Interface zur Modifikation der XML Docs wird in ***iptables::process()*** im Modul process_iptables bereitgestellt. Siehe hierzu die Kommandozeilenparameter ***firewall config-tool*** der firewall Applikation.

Services

Schema Aufbau:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
		xmlns:f="http://www.wago.com/security/firewall"
		xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:include href="../transform.xsl"/>

<xsl:output method="text" indent="no" encoding="utf-8" media-type="text/plain"/>
<xsl:strip-space elements="*" />

<xsl:template match="/f:firewall">
<xsl:apply-templates select="f:ipv4/f:service[@name]"/>
<xsl:value-of select="$newline"/>
</xsl:template>

<xsl:template match="f:ipv4/f:service[@name]">
<xsl:variable name="srv_name" select="@name"/>
<xsl:text>*filter</xsl:text>
<xsl:value-of select="$newline"/>
<xsl:text>:in_</xsl:text><xsl:value-of select="@name"/><xsl:text> - [0:0]</xsl:text>
<xsl:value-of select="$newline"/>
<xsl:apply-templates select="f:interfaces/f:interface[@state='on']">
   <xsl:with-param name="srv_name" select="$srv_name"/>
</xsl:apply-templates>
<xsl:text>-A in_services -j in_</xsl:text><xsl:value-of select="@name"/>
<xsl:value-of select="$newline"/>
<xsl:text>COMMIT</xsl:text>
</xsl:template>

<xsl:template match="f:interfaces/f:interface[@state='on']">
   <xsl:param name="srv_name"/>

   <xsl:variable name="el" select="current()"/>
   <xsl:for-each select="$parameters/f:firewall/f:parameters/f:interfaces/f:interface[@name=$el/@if]">
      <xsl:variable name="if">
         <xsl:call-template name="ifname-ipsec-in-cur"/>
      </xsl:variable>
      <xsl:apply-templates select="$el/../../f:rules/f:rule[@state='on' and @proto and @dst_port]">
         <xsl:with-param name="srv_name" select="$srv_name"/>
         <xsl:with-param name="if" select="$if"/>
      </xsl:apply-templates>
   </xsl:for-each>
</xsl:template>

<xsl:template name="service-input-filter">
   <xsl:param name="srv_name"/>
   <xsl:param name="if"/>
   <xsl:param name="proto"/>

   <xsl:text>-A in_</xsl:text>
   <xsl:value-of select="$srv_name"/>

   <xsl:text> </xsl:text>
   <xsl:value-of select="$if"/>

   <xsl:text> -p </xsl:text>
   <xsl:value-of select="$proto"/>

   <xsl:if test="@src_port">
      <xsl:text> --sport </xsl:text>
      <xsl:value-of select="@src_port"/>
   </xsl:if>

   <xsl:text> --dport </xsl:text>
   <xsl:value-of select="@dst_port"/>

   <xsl:text> -j ACCEPT</xsl:text>
```

```
   <xsl:value-of select="$newline"/>
</xsl:template>

<xsl:template match="f:rules/f:rule[@state='on' and @proto and @dst_port]">
   <xsl:param name="srv_name"/>
   <xsl:param name="if"/>

   <xsl:if test="@proto='tcp' or @proto='udp'">
      <xsl:call-template name="service-input-filter">
         <xsl:with-param name="srv_name" select="$srv_name"/>
         <xsl:with-param name="if" select="$if"/>
         <xsl:with-param name="proto" select="@proto"/>
      </xsl:call-template>
   </xsl:if>
   <xsl:if test="@proto='tcpudp'">
      <xsl:call-template name="service-input-filter">
         <xsl:with-param name="srv_name" select="$srv_name"/>
         <xsl:with-param name="if" select="$if"/>
         <xsl:with-param name="proto">tcp</xsl:with-param>
      </xsl:call-template>
      <xsl:call-template name="service-input-filter">
         <xsl:with-param name="srv_name" select="$srv_name"/>
         <xsl:with-param name="if" select="$if"/>
         <xsl:with-param name="proto">udp</xsl:with-param>
      </xsl:call-template>
   </xsl:if>
</xsl:template>

</xsl:stylesheet>
```

XML Dokument

```
<?xml version="1.0" encoding="utf-8"?>
<firewall xmlns="http://www.wago.com/security/firewall"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.wago.com/security/firewall service.xsd">
   <ipv4>
      <service name="ssh">
         <interfaces>
            <interface state="on"  if="br0"/>
            <interface state="on"  if="br1"/>
            <interface state="on"  if="br2"/>
            <interface state="on"  if="br3"/>
            <interface state="off" if="WAN"/>
            <interface state="on"  if="VPN"/>
         </interfaces>
         <rules>
            <rule state="on" proto="tcp" dst_port="22"/>
         </rules>
      </service>
   </ipv4>
</firewall>
```

Aufruf der firewall Applikation

/usr/bin/firewall {service} up|down

Konvertierung von XML Docs zu iptables

/usr/bin/xmlstarlet tr {XML Schema} {XML Doc service} > {rule File}

Anwenden der iptables Regeln

/sbin/iptables-restore -n > /dev/null 2>&1 < {rule File}

Beispiel ssh up

/usr/bin/firewall ssh up
/usr/bin/xmlstarlet tr /etc/firewall/services/service_up.xsl /etc/firewall/services/ssh.xml
> etc/firewall/services/ssh_up.rls

/sbin/iptables-restore -n > /dev/null 2>&1 < etc/firewall/services/ssh_up.rls

Produzierte Regeln für Up und down:

```
*filter
:in_ssh - [0:0]
-A in_ssh -i br0 -p tcp --dport 22 -j ACCEPT
-A in_ssh -i br1 -p tcp --dport 22 -j ACCEPT
-A in_ssh -i br2 -p tcp --dport 22 -j ACCEPT
-A in_ssh -i br3 -p tcp --dport 22 -j ACCEPT
-A in_ssh -i wwan0 -m policy --dir in --pol ipsec --proto esp --mode tunnel -p tcp --dport 22 -j ACCEPT
-A in_ssh -i tun+ -p tcp --dport 22 -j ACCEPT
-A in_ssh -i tap+ -p tcp --dport 22 -j ACCEPT
-A in_services -j in_ssh
COMMIT
```

```
*filter
-F in_ssh
-D in_services -j in_ssh
-X in_ssh
COMMIT
```

Ein Interface zur Modifikation der XML Docs wird in ***services::process()*** im Modul process_services bereitgestellt. Siehe hierzu die Kommandozeilenparameter ***firewall config-tool*** der firewall Applikation.

Packet Filter

Packet Filer befinden sich im Transport- und Network Layer.

- Auswertung der Header Pakete: zustandslos, jedes Paket einzeln
- Stateful Inspection: Beziehung zwischen Paketen auswerten
-

Schutz vor:

- SYN-Flooding (SYN-Cookies)
- fehlerhaften Paketen (z. B. widersprüchliche TCP-Flags wie SYN-Bits, ACK-Bits und Sequenznummern)
- Ping of Death, Smurf Angriffe, Teardrop Attacken, Land-Attacken
-
-

SW Initialisation

Reihenfolge Initialisierung: siehe /etc/rc.d

S01_firewall -> ../init.d/firewall

Script:

/etc/init.d/firewall
Usage: /etc/init.d/firewall {stop|start|restart|reload|force-reload}

Binary:

/usr/bin//firewall
/etc/firewall/firewall ???
/etc/config-tools/firewall -> /usr/bin//firewall

Files:

/etc/firewall/firewall.conf -> FIREWALL_GENERAL_STATE=***disabled*** oder ***enabled***

Beschreibung:

- **stop:** stop firewall framework

disable_firewall
Required: networking ifupdown

- **start:** start firewall framework during booting
  $FIREWALL_GENERAL_STATE ? enable_firewall : disable_firewall
  Required: networking ifupdown ; mountkernfs $local_fs ebtables
- **restart:** restart firewall framework after changing firewalls settings (turn on/off a service, etc)
  $FIREWALL_GENERAL_STATE ? enable_firewall enable_firewall_service : disable_firewall
- **reload:** reload firewall framework after changing firewalls settings (turn on/off a service, etc)
  $FIREWALL_GENERAL_STATE ? enable_firewall enable_firewall_service : disable_firewall
- **force-reload:** reload firewall framework after changing firewalls settings (turn on/off a service, etc)
  $FIREWALL_GENERAL_STATE ? enable_firewall enable_firewall_service : disable_firewall

SW Konfiguration

Script:

/etc/config-tools/firewall_apply.sh

Wrapper for firewall config-tool with automatic service state detection (up|down).
Usage is identical as in the case of the firewall config-tool, with the few changes:
- it works only for services! (firewall, iptables and ebtables are not allowed)
- --apply up|down option is not allowed
- it automatically detects state of the service (enabled|disabled)
- if the service is disabled it simply apply all given options as the are
- if the service is enabled it applys all given options plus --apply up at the end

z.B. */etc/config-tools/firewall_apply.sh ssh*
ruft
sudo /etc/config-tools/firewall ssh --apply up

Binary:

/usr/bin//firewall
/etc/firewall/firewall ??
/etc/config-tools/firewall -> /usr/bin//firewall

Files:

/etc/firewall/firewall.conf -> FIREWALL_GENERAL_STATE=***disabled*** oder ***enabled***

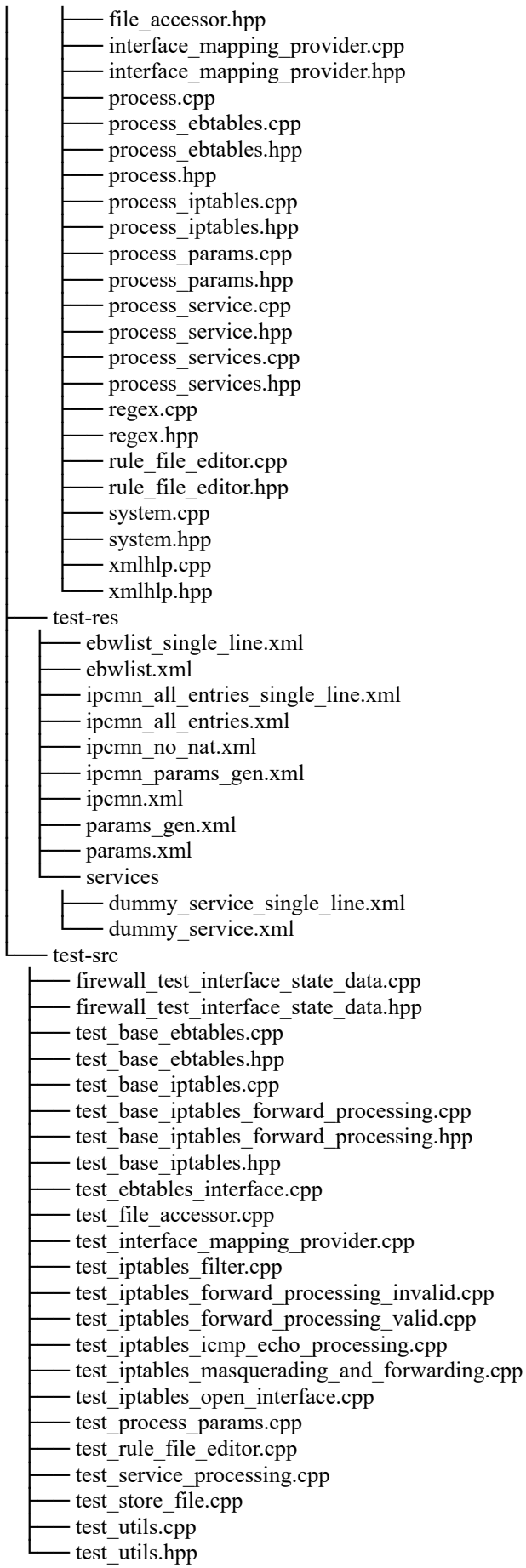
Beschreibung:

PTXDIST Package firewall-config

PTXDIST Package Source: *pfc/ptxproj/local_src/config-tools/firewall*

Namespace ist durchgängig wago::firewall

```
├── Makefile
├── make-wrapper
├── src
│   ├── app
│   │   └── main.cpp
│   └── lib
│       ├── backup.cpp
│       ├── backup.hpp
│       ├── error.cpp
│       ├── error.hpp
│       └── file_accessor.cpp
```


file_accessor.hpp
interface_mapping_provider.cpp
interface_mapping_provider.hpp
process.cpp
process_ebtables.cpp
process_ebtables.hpp
process.hpp
process_iptables.cpp
process_iptables.hpp
process_params.cpp
process_params.hpp
process_service.cpp
process_service.hpp
process_services.cpp
process_services.hpp
regex.cpp
regex.hpp
rule_file_editor.cpp
rule_file_editor.hpp
system.cpp
system.hpp
xmlhlp.cpp
xmlhlp.hpp
test-res
ebwlist_single_line.xml
ebwlist.xml
ipcmn_all_entries_single_line.xml
ipcmn_all_entries.xml
ipcmn_no_nat.xml
ipcmn_params_gen.xml
ipcmn.xml
params_gen.xml
params.xml
services
dummy_service_single_line.xml
dummy_service.xml
test-src
firewall_test_interface_state_data.cpp
firewall_test_interface_state_data.hpp
test_base_ebtables.cpp
test_base_ebtables.hpp
test_base_iptables.cpp
test_base_iptables_forward_processing.cpp
test_base_iptables_forward_processing.hpp
test_base_iptables.hpp
test_ebtables_interface.cpp
test_file_accessor.cpp
test_interface_mapping_provider.cpp
test_iptables_filter.cpp
test_iptables_forward_processing_invalid.cpp
test_iptables_forward_processing_valid.cpp
test_iptables_icmp_echo_processing.cpp
test_iptables_masquerading_and_forwarding.cpp
test_iptables_open_interface.cpp
test_process_params.cpp
test_rule_file_editor.cpp
test_service_processing.cpp
test_store_file.cpp
test_utils.cpp
test_utils.hpp

Funktion app/main.cpp

Aufrufparamerter:

```
Usage: firewall -h|--help
       firewall CONF --get-xml
       firewall CONF --apply [up|down]
       firewall CONF [--stdio] CMD [OPTIONS] [--apply [up|down]]

  -h, --help                prints this help

 Configuration type (CONF) determines allowed commands (CMD) and their options
 (OPTIONS). The following configurations are allowed:

  firewall                  general firewall settings
  ebtables                  link layer, ethernet settings
  iptables                  network layer, IPv4 settings
  services                  configuration files of all services
  SERVICE                   a standard service settings, e.g. ftp, ssh, http,
etc.

 Each option which is marked with |- sequence at its end, e.g. LIMIT|-, is opti-
onal
 and instead of real value '-' character can be set in its place. This means
that
 the option should not be used, and it will get erased from the configuration.
 Please note that whenever BURST option is used it is only allowed if its LIMIT
 counterpart is also present, i.e. is not set to '-' value.
 For exact definitions of allowed values please refer //fiewall/patterns.xsd
file.
 In some case the name of an option directly translates into a name of a rule
 with the following exceptions:
 - TOTAL     -> conn_count
 - TCP/UDP   -> limit (for limits)
 - STATE     -> onoff
 - INTERFACE -> ifname
 - MAC       -> ifmac
 - MASK      -> ifmac_mask (for mac addresses)
 - *-IP      -> ip4
 - *-PORT    -> port

 Common command:
  --get-xml                 returns chosen configuration in legacy xml form
  --get-xml-ng              returns chosen configuration in wbn-ng xml form
                            This command is preliminary. It may be changed or
                            removed in future versions.
  --apply [up|down]         applies current configuration. Please note that
                            setting values doesn't mean their immediate applica-
tion.
                            Several changes can be done to the configuration be-
fore
                            whole set is applied.
                            Options 'up' and 'down' may be used only for ser-
vices,
                            where 'up' means application of service related ru-
les
                            and 'down' their removal.

 Common option:
  --stdio                   toggles configuration source to the standard input
                            and prints results on the standard ouput. Doesn't
                            work with --get-xml and --apply commands. By default
                            configuration is taken from a standard, predefined
                            location and in place edited
```

```
 CONF: firewall:
  --is-enabled              returns 'enabled' or 'disabled' depending on the
state
                            of the entire firewall.
  --enable                  enables firewall.
  --disable                 disables entire firewall.
  --backup                  prints on stdio configuration of the entire firewall
                            in format compatible with the standard backup tool.
  --restore                 restores configuration of the entire firewall from
                            the stdio.

 CONF: ebtables:
  --set-mode MODE           set mode of work: all-allow|whitelist

  --set-log STATE TAG LIMIT|- BURST|- LEVEL
                            set logging

  --set-if STATE INTERFACE  change interface state, fully open or filtered.
                            If there is no existing entry for an interface it is
                            by default considered closed. Setting a state for
                            an interface previously not on the list adds it to
                            the list
  --rem-if INTERFACE        remove an interface from the list altogether. This
                            will have the same effect as setting its state to
                            'filtered'.

  --toggle-eproto STATE     turn on/off filtering of protocols

  --add-eproto EPROTO       add protocol to the allowed list
  --rem-eproto EPROTO       remove protocols from the allowed list

  --add-host STATE MAC MASK|-
                            add a new whitelist entry
  --upd-host INDEX STATE MAC MASK|-
                            update an existing whitelist entry
  --rem-host INDEX          remove an existing whitelist entry

 CONF: iptables:
  --set-climits TOTAL|- LIMIT|- BURST|- TCP|- UDP|-
                            sets limitations on incoming connections

  --set-echo POLICY LIMIT|- BURST|- BROADCAST_PROTECTION
                            sets global ping policy and limitations, for all
                            interfaces. BROADCAST_PROTECTION is an on/off
                            parameter, active indeptendently of global policy.
                            It triggers global echo broadcat protection
                            implemented the Linux kernel

  --set-echo-if POLICY INTERFACE LIMIT|- BURST|-
                            sets a policy and limitation for ping requests for
                            a given interface. Please note that these policies
                            are active only if default policy for all interfaces
                            is set to drop
  --rem-echo-if INTERFACE   removes an existing policy and limitation

  --set-forward STATE       toggles global forwarding, between all interfaces
  --set-forward-link STATE INTERFACE INTERFACE
                            add/update forwarding rule for two interfaces. This
                            rule will have an effect only if global forwarding
                            is disabled (see --set-forward option)
  --rem-forward-link INTERFACE INTERFACE
                            remove forwarding rule
```

```
  --set-masq INTERFACE      applies masquerading to a given interface
  --rem-masq INTERFACE      removes an existing masquerading setting
  --rem-masq all            removes all existing masquerading settings

  --add-pfw STATE INTERFACE|- PROTOCOL DEST-IP|- DEST-PORT|- FW-IP|- FW-PORT|-
                            adds a port forwarding rule. For each of
                            the following pairs one of values must be present:
                            - DST-IP or DST-PORT
                            - FW-IP or FW-PORT
  --upd-pfw INDEX STATE INTERFACE|- PROTOCOL DEST-IP|- DEST-PORT|- FW-IP|- FW-
PORT|-
                            updates an existing port forwarding rule
  --rem-pfw INDEX           removes an existing port forwarding rule
  --rem-pfw all             removes all existing port forwarding rules

  --set-open-if STATE INTERFACE
                            add/update a fully openned interface entry
  --rem-open-if INTERFACE   remove a fully openned interface entry

  --add-filter STATE INTERFACE|- PROTOCOL SRC-IP|- SRC-MASK|- SRC-PORT|- DEST-
IP|- DEST-MASK|- DEST-PORT|- POLICY
                            adds a filtering rule. At least one of the optional
                            parameters must be present. If SRC-PORT or DST-PORT
                            is given PROTOCOL also must be set. *-MASK can be
                            set only if corresponding *-IP is set as well
  --upd-filter INDEX STATE INTERFACE|- PROTOCOL SRC-IP|- SRC-MASK|- SRC-PORT|-
DEST-IP|- DEST-MASK|- DEST-PORT|-
                            updates an existing filtering rule
  --rem-filter INDEX        removes an existing filtering rule

 CONF: services:
  --get-ifs-status FORMAT   returns a summary of status of all services on
                            all interfaces. FORMAT denotes requested output for-
mat
                            and may take one of the two values: xml or json.
  --get-ifs-status-ng FORMAT returns a summary of status of all services on
                            all bridges and interfaces. FORMAT denotes requested
output format
                            and may take one of the two values: xml or json.
                            This command is preliminary. It may be changed or
                            removed in future versions.

 CONF: SERVICE:
  --set-if STATE INTERFACE  enables/disables application of service filtering
                            rules to a given interface. If an interface is not
                            on the list of enabled/disabled interfaces rules are
                            not applied to it.
  --rem-if INTERFACE        removes interface entry

  --add-rule STATE PROTO SRC-PORT|- DST-PORT
                            adds a new rule
  --upd-rule INDEX STATE PROTO SRC-PORT|- DST-PORT
                            updates an existing rule
  --rem-rule INDEX          removes an existing rule
```

Beschreibung:

Der komplette Ablauf wird in app/main.cpp abgebildet.

arg[1]	arg[2]	arg[3]	Aktion 1	Aktion 2	Aktion 3
firewall	--is-enabled		output state		
	--enable		update net-	set firewall.conf enab-	sh /etc/init.d/firewall restart

	--disable		work_interface_name_mapping	led/disabled FIREWALL_GENERAL_STATE=	sh /etc/init.d/firewall stop
	--backup		perform_backup()		
	--restzore		perform_restore()		
services	--get-ifs-status	FORMAT	process_services()		
	--get-ifs-status-ng	FORMAT	process_services()		
„X“	--get-xml		print_file „X“		
„X“	--get-xml-ng		print_file_ng „X“		
iptables	--apply	up/down /LEER	update_network_interface_name_mapping	command apply iptables	siehe sh script
ebtables	--apply			command apply ebtables	siehe sh script
iptables	--stdio	--apply ??	read configuration	iptables::process	store configuration optional appl
ebtables				ebtables::process	
??				iservice::process	

Funktion lib/backup.cpp

Backup/Restore the firewall

- sh /etc/firewall/fwbackup.sh
- restore iptables, ebtables, service

Funktion lib/error.cpp

Base: class execution_error : public std::runtime_error
Log: log_error_message
classes:

- execution_error
- unknown_error
- missing_param_error
- invalid_param_error
- file_open_error
- file_write_error
- file_read_error
- file_close_error
- system_call_error
- invalid_config_error

Funktion lib/file_accessor.cpp

class FileAccessor

- get_config_fname : Returns default path name to configuration file
- print_file : Output contains interfaces
- print_file_ng : Output contains bridges
- read_configuration : Reads and parses xml file
- store_configuration : Stores xml document
- copy_file
- check_file

Funktion lib/interface_mapping_provider.cpp

class InterfaceMappingProvider
Maps Interface X1, X2 to br0, br1

- get_interface

Funktion lib/process.cpp

- get_ctx : Creates xpath contest for a given xml document
- is_match_std : Marker wrapper for regex::is_match function
- is_match_opt : Wrapper for regex::is_match function
- updrem_attribute : Updates or removes an attribute based on a new supplied value
- remove : Removes an entry in the configuration file based on supplied parameters

Funktion lib/process_ebtables.cpp

Ein Interface zur Modifikation der XML Docs wird hiers bereitgestellt. Siehe hierzu die Kommandozeilenparameter ***firewall config-tool*** der firewall Applikation.

Namespace:+ ebtables

- process : Process ebtables's configuration change request
- impl::set_if
- impl::rem_if

Funktion lib/process_iptables.cpp

Ein Interface zur Modifikation der XML Docs wird hiers bereitgestellt. Siehe hierzu die Kommandozeilenparameter ***firewall config-tool*** der firewall Applikation.

Namespace:+ ipbtables

- process : Process iptables's configuration change request
- impl::set_echo_if
- impl::rem_echo_if
- impl::set_forward_link
- impl::rem_forward_link
- impl::set_masq
- impl::rem_masq
- impl::add_pfw
- impl::upd_pfw
- impl::rem_pfw
- impl::set_open_if
- impl::rem_open_if
- impl::add_filter
- impl::upd_filter

Funktion lib/process_params.cpp

- update_network_interface_name_mapping : Update params_gen.xml file, defining network device name mappings, e.g. ethX1 <-> br0

Funktion lib/process_service.cpp

Ein Interface zur Modifikation der XML Docs wird hiers bereitgestellt. Siehe hierzu die Kommandozeilenparameter ***firewall config-tool*** der firewall Applikation.

Namespace:+ service

- process : Process a service's configuration change request
- impl::set_if
- impl::rem_if

Funktion lib/process_services.cpp

- process_services : Process requests aimed at all services

Funktion lib/regex.cpp

Helper variables and funtions for regular expressions processing
Namespace:+ regex

- class regex
- class regexs : Class containing all regexes in compiled form
- class match_info
- is_match : Checks regex's match agains a given line
- get_match : Returns single matched part of a given line

Funktion lib/rule_file_editor.cpp

class RuleFileEditor

- remove_duplicate_lines

Funktion lib/system.cpp

- exe_cmd : Executes an external command (can be shell call)

Funktion lib/xmlhlp.cpp

class xmldoc

- get
- release
- is_empty

Apply Configuration, IpTables, EbTables und Services

Die Konfiguration ist im wesentlichen von den Kommandoargumenten 1 = conf , 2 = cmd und 3 updown abhängig.
Es wird mit der Funktion *update_network_interface_name_mapping* (lib/process_params.cpp) gestartet

conf	cmd	updown	Aktion	Aktion	Aktion
iptables	--apply		(1)		
ebtables			(2)		
‚X‘		up	(3)	(5)(rule_down)	(5)(rule_up)
		down	(4)	(5)(rule_down)	
firewall	--enable		(6)(enabled)	sh /etc/init.d/fire-wall restart	
	--disable		(6)(disabled)	sh /etc/init.d/fire-wall stop	
	--backup		perform_backup		
	--restore		perform_restore		
services			(7)		
	--stdio		shift args->cmd		

iptables			read_configuration	iptables::(8)	store_configuration
ebtables				ebtables::(8)	
?				service::(8)	

1. sh /etc/firewall/ebtables/ebfirewall.sh
2. sh /etc/firewall/iptables/ipfirewall.sh --apply
3. transform_xmldoc(/etc/firewall/services/service_up.xsl,,/etc/firewall/services/'X'.xml,/etc/firewall/services/'X'_up.rls)
4. transform_xmldoc(/etc/firewall/services/service_down.xsl,,/etc/firewall/services/'X'.xml,/etc/firewall/services/'X'_down.rls) siehe /usr/bin/xmlstarlet tr {XML Schema} {XML File} > {Outfile}
5. /sbin/iptables-restore -n > /dev/null 2>&1 < ,rule'
6. FIREWALL_GENERAL_STATE=(enabled/disabled)
7. process_services(cmd,path(services),arg[3]
8. process(xmldoc,cmd,args)

Backup / Restore

/etc/firewall/fwbackup.sh und ***/etc/firewall/fwrestore.sh***

PTXDIST Packages

- firewall-config
- wbm-ng-plugin-firewall

Target Organisation

/etc/firewall/
- transform.xsl
- fwrestore.sh
- permissions.sh
- fwbackup.sh
- templates
  - service.xml ebwlist.xml README.txt ipcmn.xml
- params.xml
- patterns.xsd
- services
  - https.xml ftp.xml iec60870_5_104.xml service_up.xsl ftps.xml iec61850_mms.xml codesysr.xml servicebkp.xsl service_down.xsl modbus_udp.xml bacnet.xml service.xsd dnp3.xml profinet.xml http.xml dns.xml codesysw.xml iocheck.xml ssh.xml bootp.xml tftp.xml snmp.xml opcua.xml ssh_up.rls modbus_tcp.xml dhcpd.xml snmps.xml
- params.xsd
- params_gen.xml
- firewall
- firewall.conf
- ebtables
  - ebwlist.xsl ebwbkp.xsl ebwlist.aa.rls ebwlist.xml ebfirewall.sh ebwlist.xsd
- iptables
  - ipcmn.aa.rls ipcmn.rls ipbkp.xsl ipcmn.xsd ipcmn.xml ipcmn.xsl ipbase.rls ipfirewall.sh ipnat.xsl
- validate_if.sh
- ipsecfirewall.sh

/sbin/iptables-restore

Usage: iptables-restore [-c] [-v] [-V] [-t] [-h] [-n] [-w secs] [-W usecs] [-T table] [-M command]
[--counters]

[--verbose]
[--version]
[--test]
[--help]
[--noflush]
[--wait=<seconds>
[--wait-interval=<usecs>
[--table=<TABLE>]
[--modprobe=<command>]

Ist ein Link zu xtables-legacy-multi
/sbin/iptables-restore -n > /dev/null 2>&1 < {rule File}

/usr/bin/xmlstarlet

Usage: /usr/bin/xmlstarlet [<options>] <command> [<cmd-options>]
where <command> is one of:
ed (or edit) - Edit/Update XML document(s)
sel (or select) - Select data or query XML document(s) (XPATH, etc)
tr (or transform) - Transform XML document(s) using XSLT
val (or validate) - Validate XML document(s) (well-formed/DTD/XSD/RelaxNG)
fo (or format) - Format XML document(s)
el (or elements) - Display element structure of XML document
c14n (or canonic) - XML canonicalization
ls (or list) - List directory as XML
esc (or escape) - Escape special XML characters
unesc (or unescape) - Unescape special XML characters
pyx (or xmln) - Convert XML into PYX format (based on ESIS - ISO 8879)
p2x (or depyx) - Convert PYX into XML
<options> are:
-q or --quiet - no error output
--doc-namespace - extract namespace bindings from input doc (default)
--no-doc-namespace - don't extract namespace bindings from input doc
--version - show version
--help - show help
Wherever file name mentioned in command help it is assumed
that URL can be used instead as well.

Type: /usr/bin/xmlstarlet <command> --help <ENTER> for command help

XMLStarlet is a command line toolkit to query/edit/check/transform
XML documents (for more information see http://xmlstar.sourceforge.net/)

Dear XMLStarlet users,
you may have noticed that the development of xmlstarlet has somewhat stalled. To get the submitted patches applied I volunteered to co-admin the project and at least do some maintenance work. Unfortunetaly my time is limited and I would like to call for participation. Especially the project needs help in the following areas:

/usr/bin/xmlstarlet tr {XML Schema} {XML File} > {Outfile}

/etc/firewall/firewall.conf

FIREWALL_GENERAL_STATE=disabled|enabled

/etc/init.d/firewall

Boot Script zum Konfigurieren der firewall

X-Start-Before: networking ifupdown
X-Stop-After: networking ifupdown
Short-Description: set iptables framework
Description: the script set's the firewall framework.
usage:
- start firewall framework during booting: *firewall start*
- stop firewall framework: *firewall stop*

- restart firewall framework after changing firewalls settings (turn on/off a service, etc):
firewall restart|reload|force-reload

Ablauf des scripts

- /etc/firewall/firewall
- /etc/firewall/firewall.conf
- start ?: enable_firewall : disable_firewall
- stop?: disable_firewall
- restart|reload|force-reload?: enable_firewall, enable_firewall_service : disable_firewall

disable_firerwall:
    ebfirewall.sh --disable
    ipfirewall.sh --disable

enable_firerwall:
    validate_if.sh
    ebfirewall.sh
    ipfirewall.sh

enable_firerwall_services:
    ipfirewall.sh --services

/etc/firewall/ebtables/ebfirewall.sh

Set link layer firewall rules
Usage:/etc/firewall/ebtables/ebfirewall.sh [--disable]

Ist nicht autonom ausführbar

/etc/firewall/iptables/ipfirewall.sh --apply

Set network layer firewall rules.
Usage: /etc/firewall/iptables/ipfirewall.s [--disable||--services]
    --disable clean firewall
    --services set_firewall_services
    --apply set_firewall $1

clean_firewall
    /usr/sbin/iptables-restore /etc/firewall/iptables/ipcmn.aa.rls
    /usr/bin/xmlstarlet ??
    set_dynamic_default ??

set_firewall_services
    RESULT=$(find /etc/config-tools/events/*/firewall -type f -exec {} start \;)

set_firewall
    /usr/sbin/iptables-restore ??
    /usr/bin/xmlstarlet ??
    ??

Ist nicht autonom ausführbar

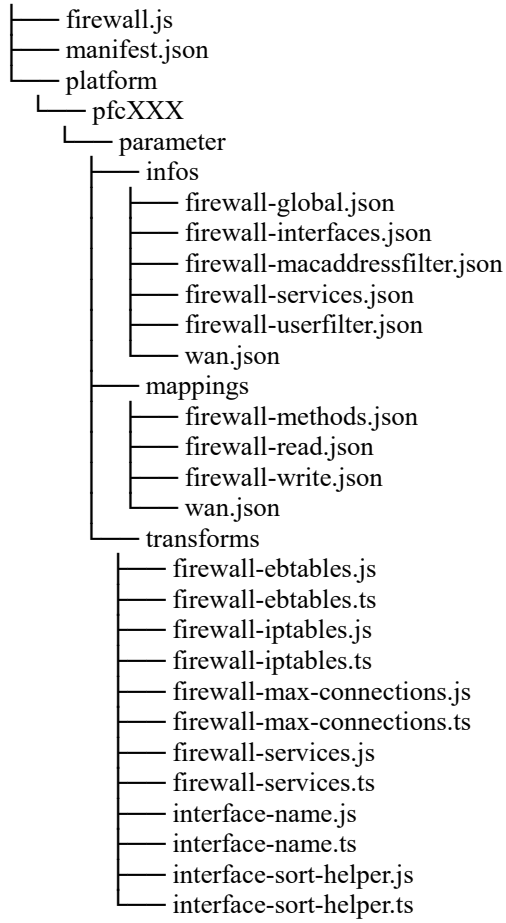
/usr/bin/firewall

Das installierte *firewall-config* Package

wbm-ng-plugin-firewall

Im Rahmen eines php-cgi Servers läuft dieses Plugin und überträgt die Firewall Konfiguration auf das oben beschriebene Tool /usr/bin/firewall.

Siehe: pfc/ptxproj/platform-wago-pfcXXX/build-target/wbm-ng-plugin-firewall



Die *.js Dateien sind komprimiert und ohne Rückwandlung nur schwer lesbar.

Beispiel: platform/pfcXXX/parameter/mappings/firewall-methods.json

```
[
    {
        "command": "firewall ebtables --add-host $state $address $mask --apply",
        "executes": ["firewall.macaddressfilter.addwhitelist"],
        "type": "mapping",
        "mapping": {
            "enabled": "state",
            "address": "address",
            "mask": "mask"
        },
        "conversion": {
            "enabled": [
                {"from": true, "to": "on"},
                {"from": false, "to": "off"}
            ]
        }
    },
    {
        "command": "firewall ebtables --rem-host $$index",
        "constants": {
            "index": "firewall.macaddressfilter.whitelist.*.index"
        },
        "executes": ["firewall.macaddressfilter.whitelist.*.delete"]
    },
    {
        "command": "firewall iptables --add-filter on $iface $protocol $sourceIp $sourceMask $sour-
cePort $destIp $destMask $destPort
 $policy --apply",
        "executes": ["firewall.adduserfilter"],
        "type": "mapping",
        "mapping": {
            "iface": "iface",
            "protocol": "protocol",
            "sourceIp": "sourceIp",
```

```
            "sourceMask": "sourceMask",
            "sourcePort": "sourcePort",
            "destIp": "destIp",
            "destMask": "destMask",
            "destPort": "destPort",
            "policy": "policy"
        },
        "conversion": {
            "iface":      [{"from": "Any", "to": "-"}],
            "sourceIp":   [{"from": "", "to": "-"}],
            "sourceMask": [{"from": "", "to": "-"}],
            "sourcePort": [{"from": "", "to": "-"}],
            "destIp":     [{"from": "", "to": "-"}],
            "destMask":   [{"from": "", "to": "-"}],
            "destPort":   [{"from": "", "to": "-"}]
        }
    },
    {
        "command": "firewall iptables --rem-filter $$index --apply",
        "constants": {
            "index": "firewall.userfilter.*.index"
        },
        "executes": ["firewall.userfilter.*.delete"]
    }
]
```

Mapping auf das Tool /usr/bin/firewall

Bestehende Dokumentation

Firewall – Wikipedia

Firewall-Regelwerk – Wikipedia

Firewalls For Dummies, 2nd Edition (lagout.org)

iptables › Wiki › ubuntuusers.de

ebtables(8) - Linux man page (die.net)

xtables-legacy(8) - Linux manual page (man7.org)

Computernetzwerke; Andrew S. Tanenbaum, David J. Wetherall Seite 924ff

Wago spezifisch

waw/pages/viewpage.action?spaceKey=EA&title=Firewall+-+PFC