CHECK POINT
INFINITY

The Cyber Security Architecture of the Future

R80.10 Next Generation Threat Prevention Platforms

Check Point®
SOFTWARE TECHNOLOGIES LTD

Learn More: checkpoint.com

THE SECURITY GATEWAY ARCHITECTURE OF THE FUTURE

Overview

The exponential growth of data from various sources like network, cloud, mobile, and virtual systems constantly introduces new threats. Organizations need to quickly adapt and protect themselves, requiring more powerful, high-performance security.

Check Point Infinity, the first consolidated security across networks, cloud and mobile, provides the highest level of threat prevention against both known and unknown targeted attacks to keep you protected now and in the future.

Check Point R80.10, part of Check Point Infinity, takes security to new levels, merging security into a single system which enables administrators to easily identify security risks across the organization.

Unified policies in R80.10 gateways enables organizations to translate their security definitions into a simple set of rules, which then streamline policy administration and enforcement throughout the organization. Policy layers provide the ability to separate the policy into independent segments, which can be independently managed and automated. In the kernel the rule base is represented as a database table. This enables the gateway security policy to automatically adapt as dynamic objects change, providing simple automated security across physical, virtual and cloud environments.

First, the Basics of Check Point NGTP Architecture

Stateful inspection, network and port address translation (NAT), Virtual Private Network (VPN) functions and forwarding are applied per-packet on the inbound and outbound interfaces of the device. State-related information is extracted from packets and maintained in dynamic state tables to evaluate subsequent connection attempts. When possible, throughput and connection rate are accelerated by a security acceleration (SecureXL) module.

Session-based processing enforces advanced access control and threat detection and prevention capabilities. To do this we assemble packets into a stream, parse the stream for relevant contexts and then security modules inspect the content. When possible, a common pattern matcher does simultaneous inspection of the content for multiple security modules. In multi-core systems this processing is distributed amongst the cores to provide near linear scalability on each additional core.

Security modules use a local cache to detect known threats. This local cache is backed up with real-time lookups of an online cloud service. The result of cloud lookups are then cached in the kernel for subsequent lookups. Cloud assist also enhances unknown threat detection and prevention. In particular a file whose signature is not known in a local cache is sent to our cloud service for processing where compute, disk and memory are virtually unlimited. Our sandboxing technology, SandBlast Threat Emulation, identifies threats in their infancy before malware has an opportunity to deploy and evade detection. Newly discovered threats are sent to the cloud database to protect other Check Point connected gateways and devices. When possible, active content is removed from files which are then sent on to the user while the emulation is done.

R80.10 Enhancements in a Nutshell

In R80.10 SecureXL adds support for Domain Objects, Dynamic Objects and Time Objects. CoreXL accelerates VPN traffic by distributing Next Generation Threat Prevention inspection across multiple cores.

The unified access control policy defines firewall, application control, URL filtering, mobile access and now content awareness in one policy rule. Content awareness adds visibility and control over data using data types based on content, file types and direction as the content transverses the network. In R80.10 you can also group interfaces of gateways into security zones in source and destination definitions.

Core IPS protections are included in the access control policy while other IPS protections are now included in the Threat Prevention policy unifying IPS, Antivirus, Anti-Bot, Threat Extraction, and Threat Emulation into one threat prevention profile. Likewise the threat prevention policy can include multiple profiles for each security gateway to apply more granular Threat Prevention policies.

Layers in the R80.10 policy are a new way to organize security. A policy can have one or more layers as its building blocks. You can use combinations of Ordered Layers, Inline Layers, and Domain Layers (in Multi-Domain environments).

Build a rule base with layers, each with a set of the security rules. Layers are inspected in the order in which they are defined, giving control over the rule base flow and precedence of security functionality. If an "Accept" action is done in a layer, inspection continues in the next layer. Sub-Policies are sets of rules that you attach to specific rules. If the rule is matched, inspection continues in the sub-policy attached to the rule. If the rule is not matched, the sub-policy is skipped. For example, a sub-policy can manage a network segment or branch office.

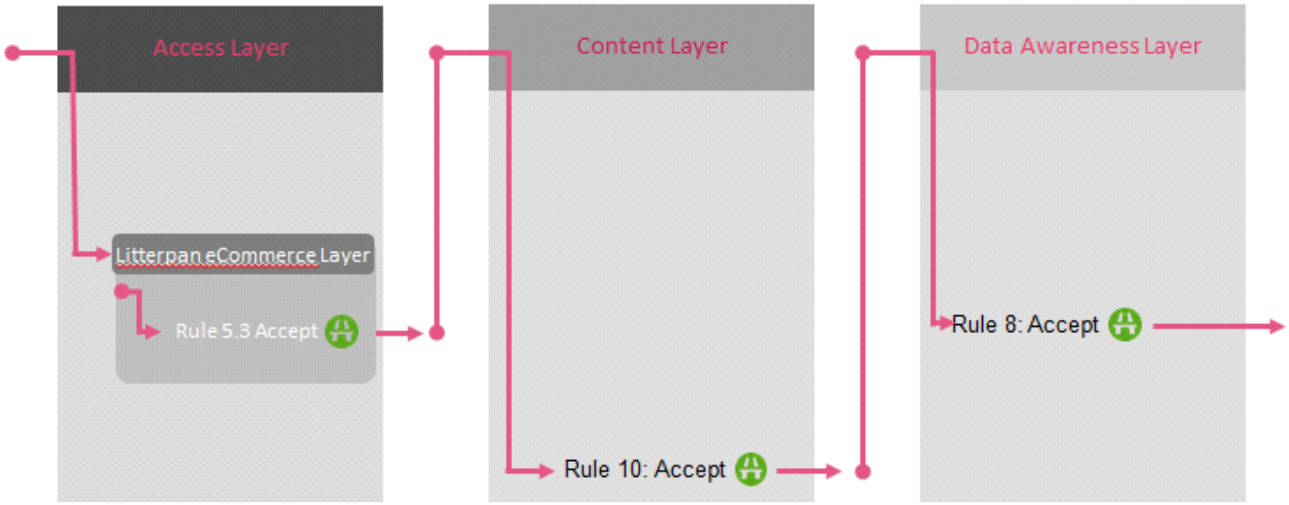


Figure 1: Policy mixed with Ordered and Inline Layers

Policy Layers and Sub-Policies can be managed by specific administrators, according to their permission profile, allowing easy responsibility delegation in the team. For more information about R80.10 management and layers check out the R80.10 Security Management Architecture whitepaper.

Services, Applications and Logs in Sync in the Unified Policy

In R77 access control is applied in an ordered fashion, for instance firewall rules are checked before application control rules. There was duplication in the firewall and application object definitions and in the policies. In R80.10 this is simplified by defining which security features you want the policy layer to inspect in the unified policy. Service objects, application objects and logging are all in sync with the policy definition.

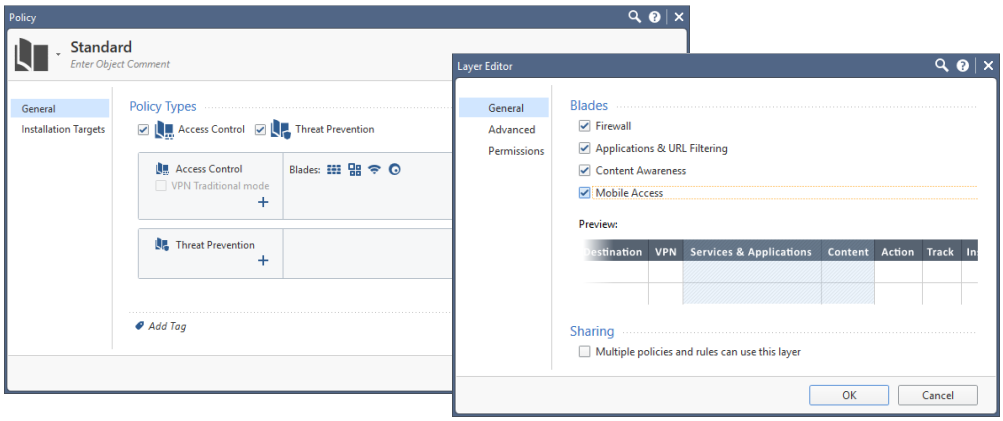


Figure 2: Unified Policy Layer Definition

Service Objects and Protocol Signatures

In R77 with separate policies for firewall and Application Control & URL Filtering there are Services Objects for the firewall policy and Network Service applications for the Application Control policy. In R80.10 this duplication is eliminated and the number of protocol handlers available in the advanced settings for the firewall service objects is reduced from over 200 to less than 20. In its place the default is to match by port in the service object and as an option to clone the service object and enable match by protocol signature. This enhances policy matching and changes the number of packets needed to define the session depending upon whether firewall does the match or application control. When application control does the match more packets are needed to make a decision about the session.

Consider the R80.10 service object SMTP which matches TCP connections where the destination port is 25. When using this object in a policy, there is no need to do transaction level analysis to identify the connection as SMTP protocol traffic. The SYN packet in the three-way TCP handshake can simply be dropped or accepted if the destination port is 25. Once the match is made the SMTP protocol handler continues to inspect the session for security issues.

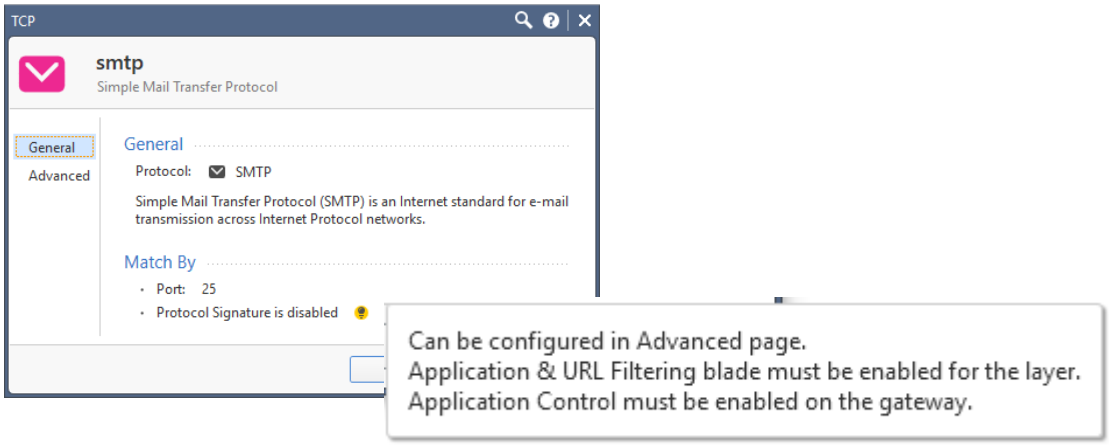


Figure 3: Default Service Object Definition

When the SMTP service object is cloned and protocol signature is enabled, Application Control must be enabled on the gateway and in the policy layer. As noted earlier and in the UI, protocol detection may allow several packets to pass to accurately identify the protocol.

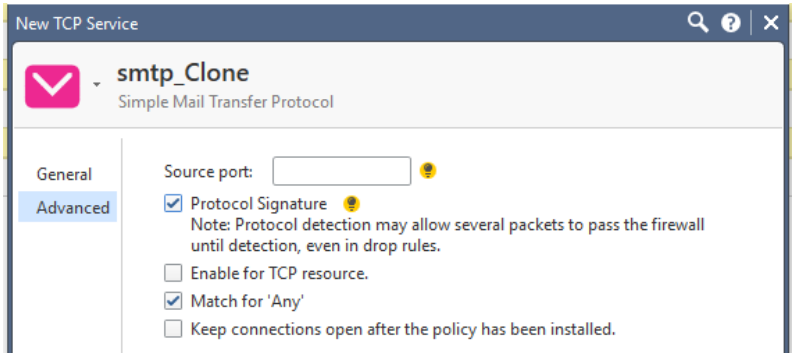
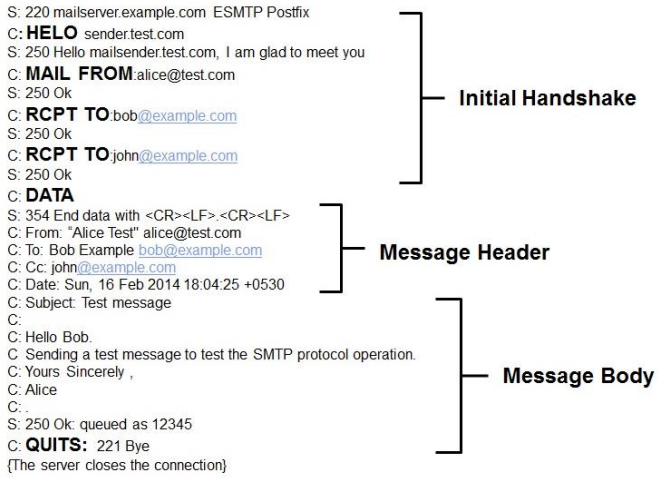


Figure 4: Cloned Service Object with Protocol Signature Match

Consider an example SMTP session and some example C2S (Client-to-Server) requests and S2C (Server-to-Client) responses. A few packets will be needed to correctly identify the protocol.



R80.10 Applications and Recommended Services

We have seen how R80.10 Service objects and application signatures are combined. Likewise, R80.10 applications now include recommended services such as HTTP and HTTPS in the application definition simplifying the unified policy configuration. For instance the R80.10 Facebook application definition includes the Web services where it is common to see Facebook traffic; HTTP/S over port 80/443 and HTTP/S proxy connections over port 8080. In the UI you can see the services included in the application definition.

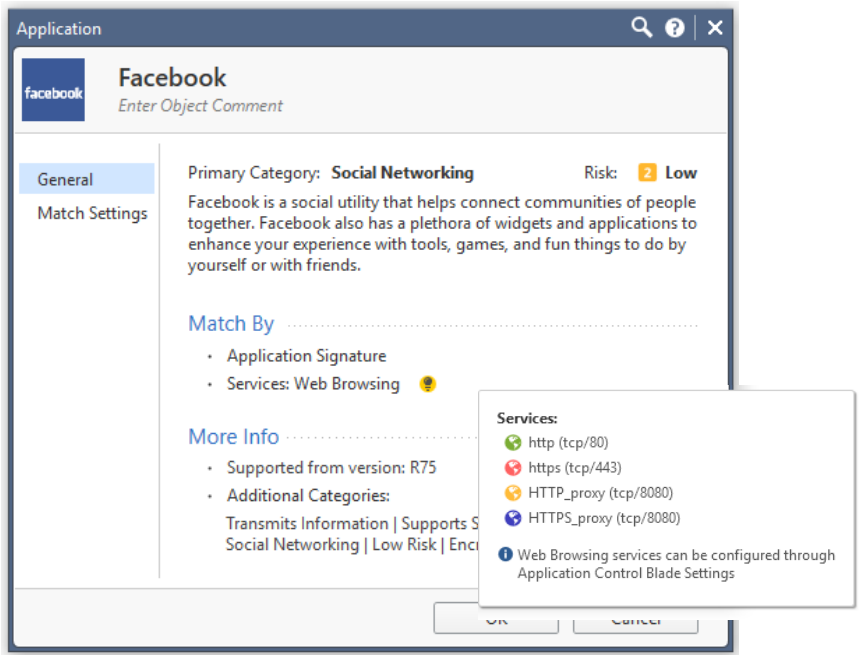


Figure 5: Facebook Application Object

Match Any Application

In R80.10 the R77 application object Any Recognized (see below) is not needed. Any in an R80.10 policy with Application Control enabled matches any application.

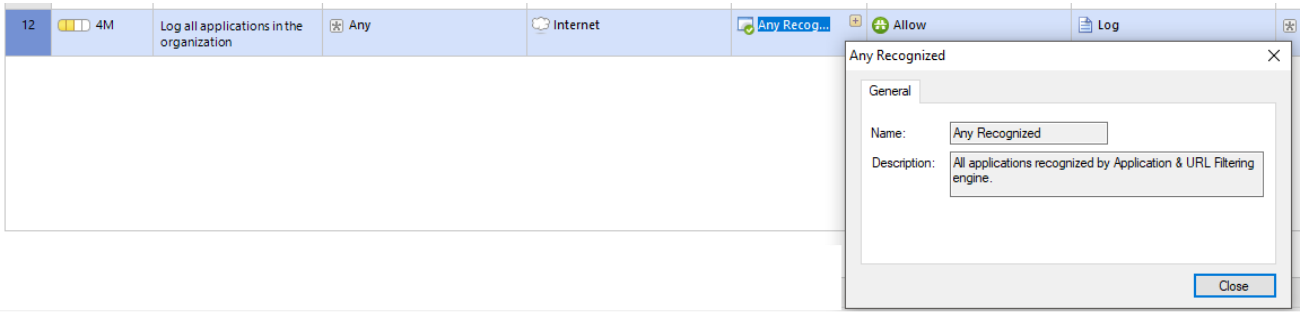


Figure 6: R77 Any Categorized Application Object

Log Options Match Policy Definition

Logging options match the policy. When only firewall is enabled on the policy the default Track option is Log. At a minimum, this is the Source, Destination, Source Port, and Destination Port. When Application Control or Content Awareness is enabled then the Detailed or Extended logging options are available. The Detailed option logs the application and the Extended option logs the full list of URLs and Files.

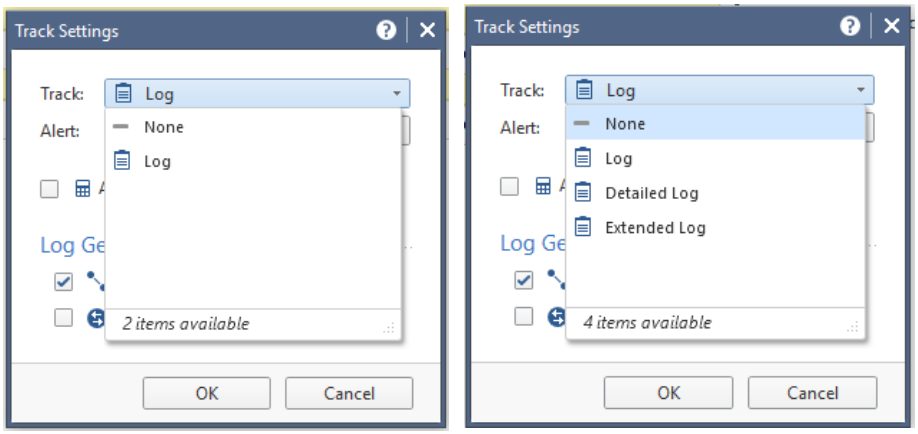


Figure 7: Firewall Policy and Application Control Policy Track Options

When an application object is added to the rule, then by default session logging is also enabled. A session is a user's activity at a specified site or with a specified application. The session starts when a user connects to an application or to a site. The Security Gateway includes all the activity that the user does in the session in one session log. Select per Session in the Track option to generate one log for all the connections in the same session. Connection unification for the logs in the session is set in the Advanced Application Control Blade settings and has a default value of 180 minutes. This is the default for rules in a Layer with Applications and URL Filtering or Content Awareness enabled in the R80.10 Unified Policy.

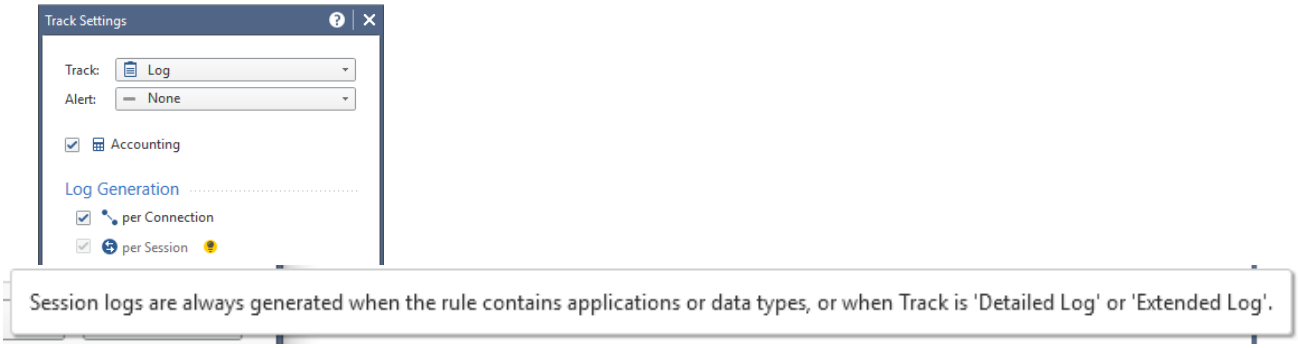


Figure 8: Aggregate Connection Logs into Session Logs

Access Control Column-based Rule Matching

Under the hood in R77 the policy matching process for Application Control, anti-malware, DLP (Data Loss Prevention and NAT (Network Address Translation) is done using a column-based search process. In R80.10 this process is now used to match the connection against the unified policy. The resulting match is still the first rule to match from the top-down. This has not changed. Only the process for finding the match has changed.

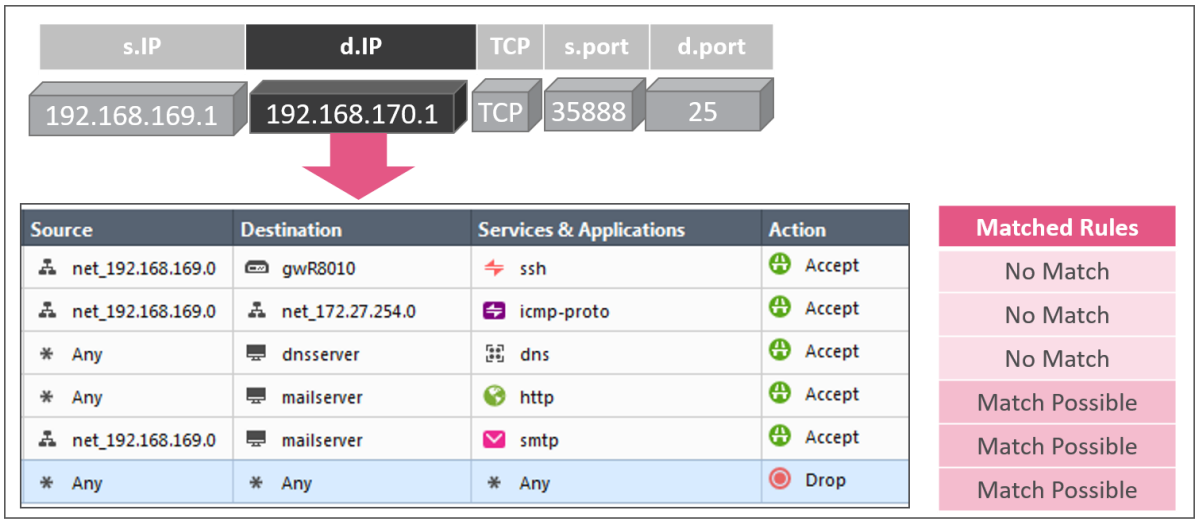
For example consider a firewall policy with service objects defined in the Services & Applications column trying to match an SMTP connection. We match the SYN packet in the three-way handshake. The search order is;

1. Destination column
2. Source column
3. Service column

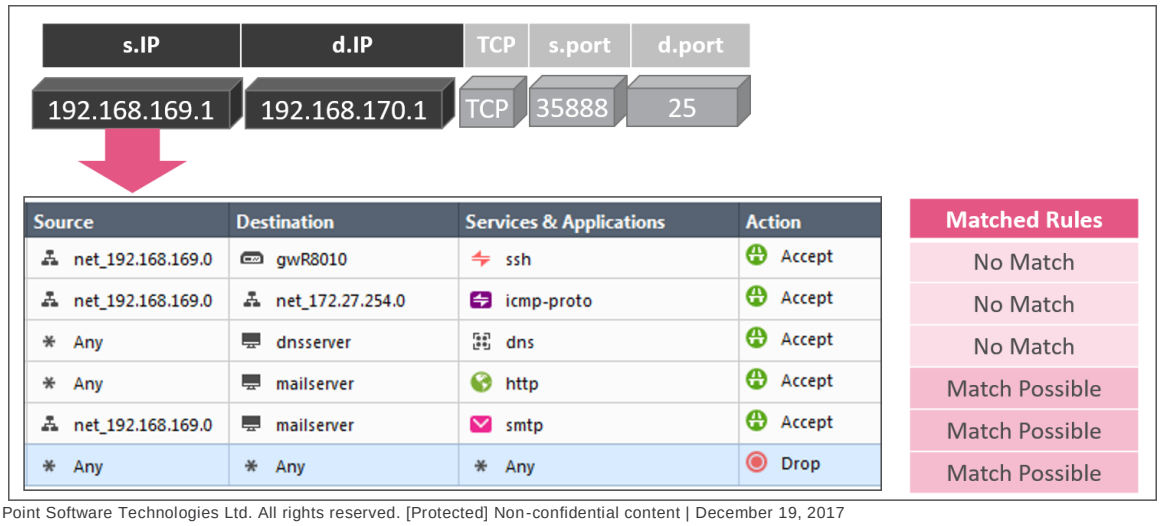
We search each column in the policy. At the end of the search we update a matched rules array. In each pass some rules can be eliminated from the matched rules array. When the rule base is large, this results in a more efficient matching process.

Consider a policy with only the firewall enabled and the rule base match of the initial SYN packet in the TCP three way handshake from a client at 192.168.169.1 connecting to the SMTP service listening on port 25 of a mail server at 192.168.170.1.

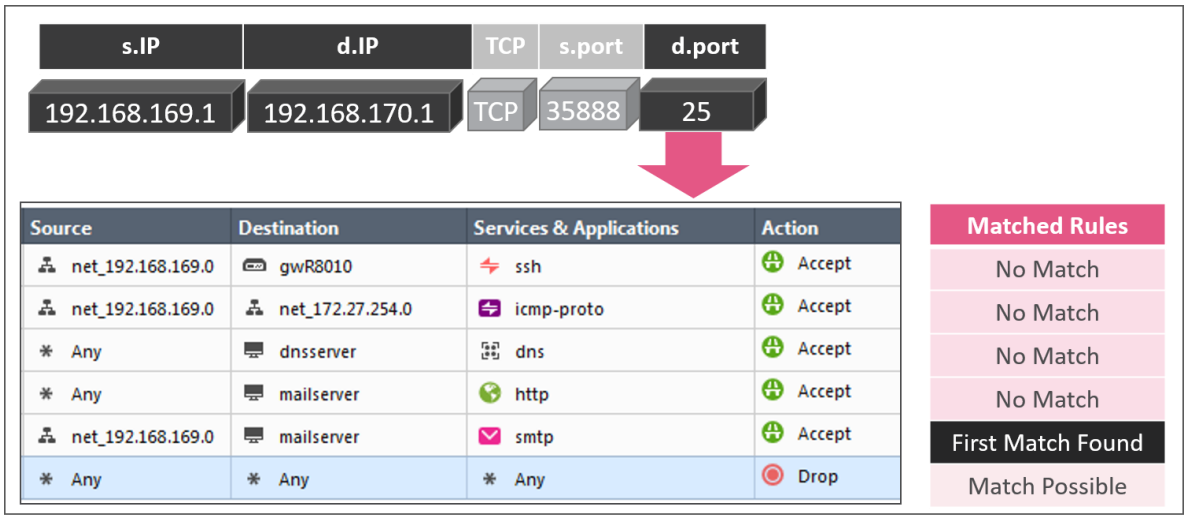
In the destination column pass of rules 1 through 6, rules 1, 2 and 3 are eliminated from the matched rules array.



In the source column pass of rules 1 through 6, rules 4 and 5 are still possible matches in the matched rules array.



In the service column pass of rules 1 through 6, rule 4 is eliminated from the matched rules array and rule 5 is a final match.



For those who are familiar with Check Point chain modules, there isn't a new Unified Policy chain module. The Unified Policy is enforced for the first packet in the VM chain module where the security rule base was enforced before. In a Unified Policy rule base with Application Control and Content Awareness enabled and a more complex policy there may not be a final match on the SYN packet. The rule base will be executed on parser contexts in subsequent packets.

Classifying Traffic to Match Policy Column Objects

In R80.10 the unified policy classifies subsequent packets against the column definitions until enough information is known to identify the traffic. Like column-based rule matching as more is known about the connection, then rules are eliminated from the matched rules array.

Example classification objects are application, access role source, access role destination, content and file.

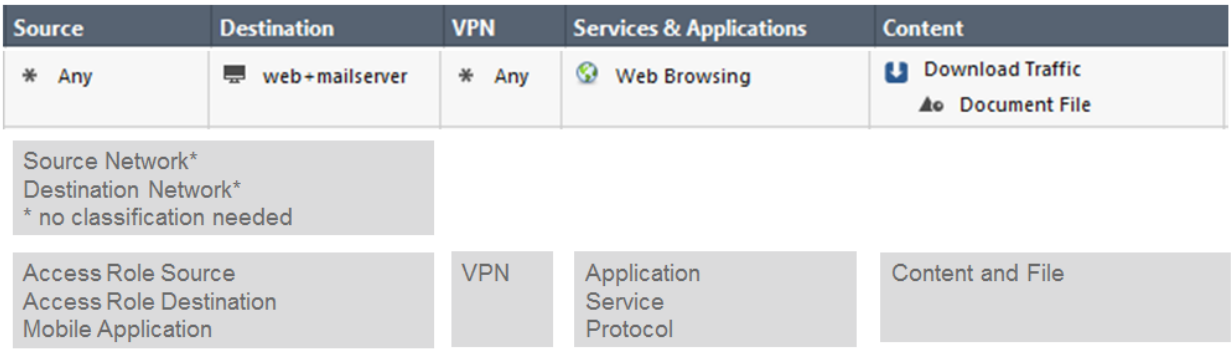


Figure 9: Classification Objects for Each Column

Consider the example rule base below that is similar to the R80.10 online help rule base matching example 3 for a connection from the InternalZone to the Internet to download an EXE file from a business web site.

Figure 10: Example Unified Policy Pass One

Classification starts with the SYN packet in the TCP three-way handshake to a Web services port. Rules 1, 2 and 3 are possible matches. The connection is allowed on rule 3 and allowed to continue. The CoreXL dispatcher notices it's a first packet and forwards the packet to the fw_worker Context Management Infrastructure (CMI). Unified policy processing begins to classify the packet by creating a list of Classifier Apps that are registered for first packets. Each Classifier App executes on the packet and creates a Classifier Object to be used to match the column definition in the security policy.

Because Application Control is enabled and more packets need to be analyzed in subsequent packets, streaming is needed. Once the TCP three-way handshake completes, then the client sends an HTTP GET to download the file. Consider the following HTTP Headers from the client request and the server response.

C: GET /download/example.exe HTTP/1.1
C: Host: www.example.com
S: HTTP/1.1 200 OK
S: Content-Type: application/octet-stream
S: Content-Length: 1136132

When checked against the security policy it is determined that application and file classification objects are needed and more packets are needed. Subsequent packets are handled by the streaming engine. The streaming engine notifies the Classifier to perform the classification. The unified processing waits for information from the CMI.

Streaming, CoreXL and SecureXL technologies have not changed. The streaming engine passes the stream to the Protocol Parser. The Protocol Parser verifies if the protocol is correct or not and provides the result to the CMI. If the protocol is correct the Parser extracts the Context. If the protocol is not correct the CMI may instruct the Parser to extract Context even then. This Context is then executed in the Pattern Matcher to identify the application. The Pattern Matcher tells the result to the CMI. For more information about these NGTP components see Check Point Security Gateway Architecture and Packet Flow (advanced access).

The CMI sends the information describing the result of the Protocol Parser and the Pattern Matcher to the Classifier. The Classification Object is checked against the rule base.

The result of the rule base check is provided to the CMI which then decides to allow or to drop the connection. The CMI generates a log message and instructs the streaming engine to forward the packets to the outbound interface.

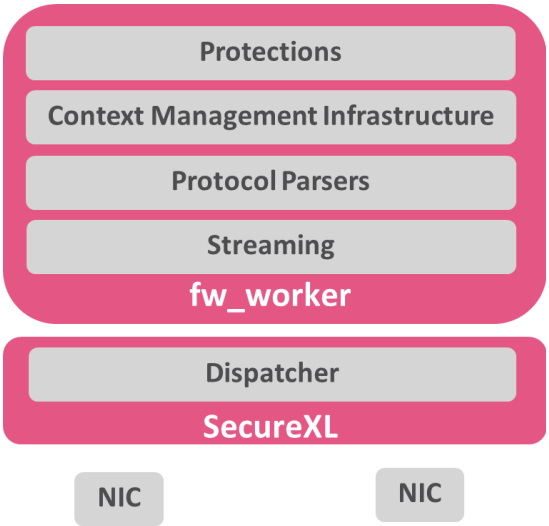


Figure 11: NGTP Architecture Components

This is the process. In our example 3, the rule base determines from the client request HTTP Header that www.example.com is not a gambling site so rule 2 can be eliminated and the client request is allowed.

Figure 12: Example Unified Policy Pass Two

Rule 1 and 3 are still a possible match. The file is extracted from the HTTP body in the server response. The Pattern Matcher determines that the file type is an EXE file. The result is sent to the classifier and checked against the rule base. Rule 1 matches. The matching process is complete and the packet is dropped.

Figure 13: Example Unified Policy Pass Three

Dynamic Updates in the R80.10 Policy

In R80.10 the rule base is represented as a database table in the kernel which automatically adapts to changes in dynamic objects and results in a more efficient policy installation process.

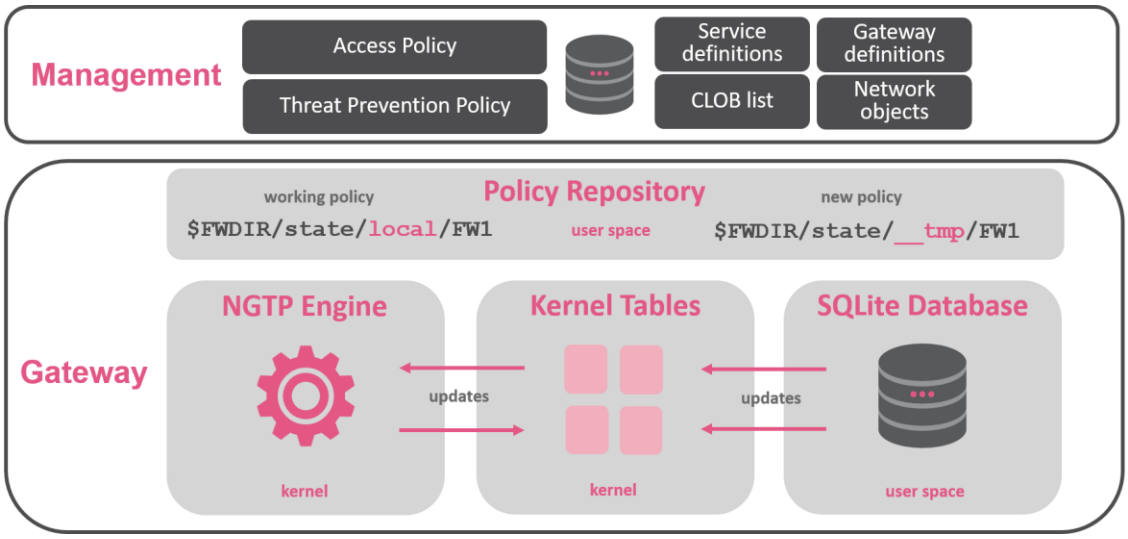


Figure 14: R80.10 Policy Installation Components

The policy installation process on R80.10 gateways consumes less time and resources than on R77 gateways. Instead of a compilation process the management server does a database scheme conversion. Each gateway receives only the objects required for the local rule base.

On the gateway a separate preparation and commit phase provides recovery and fallback if needed. The current policy is stored in the $FWDIR/state/local/FW1 directory. During the policy installation files containing the information about the new policy are transferred from the management server to the gateway and stored in the. $FWDIR/state/__tmp/FW1 directory.

The next step is to create and/or update the table structure in the SQLite database. If a failure or reboot occurs at this point, then the gateway continues to run the current policy. During the final commit stage, the pointers in kernel memory are changed to point to the new policy and objects. The gateway continues to run the current policy until the new table pointers are loaded.

In addition to supporting fallback and recovery the new structure automatically adapts to changes in dynamic objects and adds SecureXL support for Dynamic, Domain and Time objects. In R77 these objects would disable SecureXL accept templates at that point in the policy. In R80.10 security enforcement is done based upon the kernel table arrays and the policy stored in the kernel memory. As network and dynamic objects change in the SQLite database the kernel tables are updated. When the objects change the SecureXL template is revoked and then SecureXL goes through the process of recreating the connection providing acceleration template support for these dynamic objects in the R80.10 policy.

In addition to the above, the infrastructure still supports legacy commands such as “fw unloadlocal” and “fw fetch localhost” and enables future enhancements such as partial policy installs, i.e. installing a policy layer or sublayer.

Summary

Check Point R80 is the product version that brings together the security, intelligence and management capabilities to meet the many and varied IT security demands of today. While R80 is a product version, the entire, unique set of capabilities in the Check Point Infinity architecture is much more than a single product. They set the foundation for you to design and deploy a cohesive security infrastructure, a single system in fact, that will meet your security requirements now, yet is also extensible to meet your changing requirements in the future.

APPENDIX A

Unified Policy (UP) Infrastructure Components

- UP Manager of the UP infrastructure
- Classifier creates classifier objects (CLOBs) to match against the UP policy
- Observer decides if enough is known in the CLOB to check it against the rule base
- Column based rule base enforcement matches CLOBs against the policy
- Handle stores the rule base execution state

Manager

The UP Manager controls all interactions of the components and interfaces with the Context Management Infrastructure (CMI) Loader, the traffic director of the CMI (see Figure 11).

The UP Manager also has a list of Classifiers that have registered for “first packets” and uses a bitmap to instruct the UP Classifier to execute these Classifier Apps to run on the packet. The “first packets” arrive directly from the CMI. Parsing of the protocol and streaming are not needed in this stage of the connection. For “first packets” the UP Manager executes the rule base.

Classifier

When the “first packet” rule base check is complete Classifiers initiate streaming for subsequent packets in the session. The “first packet” rule base check identifies a list of rules that possibly may match and a list of CLOBs that are required to complete the rule base matching process. The Classifier reads this list and generates the required CLOBs to complete the rule base matching. Each Classifier App executes on the packet and tells the result of the CLOB to the UP Manager. The CMI then tells the Protocol Parser to enable streaming.

In some cases Classifier Apps do not require streaming, e.g. the first packet information is sufficient. Then the rule base decision can be done on the first packet.

- Dynamic Objects
- Domain Objects
- Only the firewall is enabled

On subsequent packets the Classifier can be contacted directly from the CMI using the CMI Loader infrastructure, e.g. when the Pattern Matcher has found a match it informs the CMI it has found application xyz. The CMI Loader passes this information to the Classifier. The Classifier runs the Classification Apps to generate CLOBs required for Application Control and sends the CLOBs to the Observer.

Observer

The Observer decides if enough information is known to publish a CLOB to the security policy. CLOBs are observed in the context of their transaction and the connection that the transaction belongs to. The Observer may request more CLOBs for a dedicated packet from the Classifier or decides that it has sufficient information about the packet to execute the rule base on the CLOB, e.g. if a file type is needed for Content Awareness and the gateway hasn’t yet received the S2C response containing the file. Executing the rule base on a CLOB is called “publishing a CLOB”. The Observer may wait to receive more CLOBs that belong to the same transaction before publishing the CLOBs.

Security Policy

The Security Policy receives the CLOB published by the Observer. The CLOB includes a description of the Blade it belongs to so that matching can be performed on a column basis. The security policy saves the current state on the transaction Handle; either to continue the inspection or final match.

The first packets are received directly from the UP Manager. Subsequent packets are received by the rule base from the Observer.

Handle

Each connection may consist of several transactions. Each transaction has a Handle. Each Handle contains a list of published CLOBs. The Handle holds the state of the security policy matching process. The Handle infrastructure component stores the rule base matching state related information.

Subsequent Packets

Subsequent packets are handled by the streaming engine. The streaming engine notifies the Classifier to perform the classification. The Classifier will notify the UP Manager about the performed classification and pass the CLOBs to the Observer. The CLOBs will then be received by the Observer that will need to wait for information from the CMI. The CMI sends the information describing the result of the Protocol Parser and the Pattern Matcher to the Classifier. The Classifier informs the UP Manager and sends the CLOB to the Observer. The UP Manager then instructs the Observer to publish the CLOBs to the Rule Base.

The Rule Base is executed on the CLOBs and the result is communicated to the UP Manager. The CLOBs and related Rule Base state are stored in the Handle. The UP Manager provides the result of the rule base check to the CMI that then decides to allow or to drop the connection. The CMI generates a log message and instructs the streaming engine to forward the packets to the outbound interface.

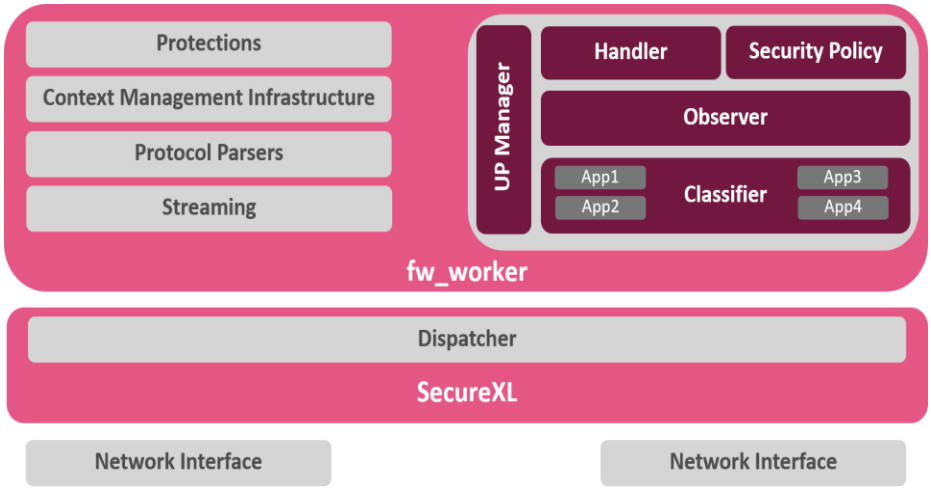


Figure 15: R80 Gateway Architecture Components