Protecting IoT (Internet of Things) implementations with R80.10 and later Unified Policy, Protocol Signature, and Segmentation

Defining protection of IoT technologies, like Ring, Ecobee, WEMO, Honeywell, Insteon, and Bloomsky, or ADT security and VOIP solutions like Ooma, in a well segmented and defined network utilizing the full suite of Check Point R80.10 and later technologies, including Protocol Signature and inline layers.

Eric Beasley (ericb@checkpoint.com)
US Channel Sales NA
2018-12-19

Table of Contents:

1 Overview and Concepts

Defining protection of IoT technologies, like Ring, Ecobee, WEMO, Honeywell, Insteon, and Bloomsky, or ADT security and VOIP solutions like Ooma, in a well segmented and defined network utilizing the full suite of Check Point R80.10 and later technologies, including Protocol Signature and inline layers.

This document will provide an overview of how to utilize Check Point Software Technologies R80.10 and later to implement security for IoT (Internet of Things) implementations in a small, but well segmented network protected by a gateway or cluster, using real world implementation examples from MyBasementCloud home production environment.

The provided treatise is suitable for adoption to larger networks by expansion of concepts related to the network and gateway approach.

SMB Appliance approach is not covered in this document, as there is currently no available R80.10 or later version of Check Point SMB appliance firmware, so key concepts like Unified Policy and Protocol Signature are not available.

1.1 Notes on example environment – MyBasementCloud Infrastructure

MyBasementCloud Infrastructure currently protects the engineer's operational home office environment and all normal network operations and access that are not explicitly for lab utilization. This infrastructure includes significant network equipment for support of layer-3 segmentation, as well as use of both 10G and 1G Ethernet. ISP services utilize enterprise commercial grade service with fixed, routable IP v4 addresses.

Established policy is continually refined and was refactored from a standard R77.30 and earlier approach for specific network and application control policy to the R80.10 level Unified Policy over an iterative and continuous process—that is still ongoing.

Some elements and specific granularity in policy is for clarification in logging, which is ubiquitously utilized to show operation and generate data for demonstration and learning.

At the time of this document's writing, the environment has a single R80.20 GA Take 101 gateway installation with R80.20 JHF (Jumbo Hotfix) 10 applied; this is still operating on the 2.6 Linux kernel. Gateway hardware is an open server installation on Supermicro SuperServer 5018D-FN8T Xeon D that is not on the Check Point Software Technologies HCL (Hardware Control List), thus operating at the Engineer's own risk. Management is operating on open server hardware, also Supermicro, similar in class and capability to the Smart-1 50, 2 CPUs with 4 cores each and currently 16 GB RAM and 1 TB HDD. Management installation is with R80.20 GA Take 101 Security Management Server installation, with JHF 10 applied and utilizes the new 3.10 kernel natively. All systems are operating in 64-bit mode and utilize a variety of Intel NICs

either integrated or added (e.g. Intel X520-DA, essentially Intel X520-2SR without the SFP+s installed).

Networking leverages different vendors, like Ubiquity, Dell, D-Link, Aruba, and HP, with WLAN APs (Access Points) from Aruba and EnGenius.

IoT (Internet of Things) solutions range from house external and WLAN connected solutions like Ring cameras and doorbell or Bloomsky weather systems, to internal solutions like Ecobee thermostat and WEMO lighting controls (via plug-in adapters). But also technologies like ADT Pulse security controller and OOMA VOIP systems for telephony.

Similar to the IoT security implementation, certain multi-media solutions, like Samsung Smart-TV, Apple TV, Roku, Sony Play Station, Microsoft XBOX, and Yamaha audio visual hardware, are handled specific to their connectivity needs, but segmented from the IoT, and also, if not needed for media sharing, segmented from other internal networks.

The provided example policy rules may at times include much more level of granularity and specific entries to cover objects and networks to allow for deeper visibility specific to the target, but also adds capabilities to get granular with controls as necessary; this may not reflect a best practice, but serves the engineers requirements for visibility and detail logging.

1.2 Definitions and Technologies

1.2.1 IoT (Internet of Things)

IoT or Internet of Things covers a growing range of technologies for enablement of a wide variety of functions, especially in environmental/home control, like Ecobee, Honeywell, or Nest thermostats for HVAC, or security/monitoring technologies like Ring cameras and doorbells, WEMO lighting/outlet controls, and home automation technologies like Insteon.

Multimedia devices with expanded functions are also an overlapping element of this technology area, especially home to Internet services interfaces like Amazon Alexa, Apple Siri, and such.

IoT technologies generally are controlled external to the home or office [or business] via an application either on the web or as a mobile device App, with data going or coming from cloud based solutions, and updates coming from the same cloud services.

1.2.2 Segmentation

Segmentation addresses the separation of network traffic to provide a walled garden environment for specific traffic and devices that includes a [security] gateway to allow access to other networks, ideally with a security context and defined rules for such access.

Segmentation can start with simple physical separation on interface ports for the gateway, or VLANs can allow a greater range of devices on an interface, while separating the traffic for specific purposes.

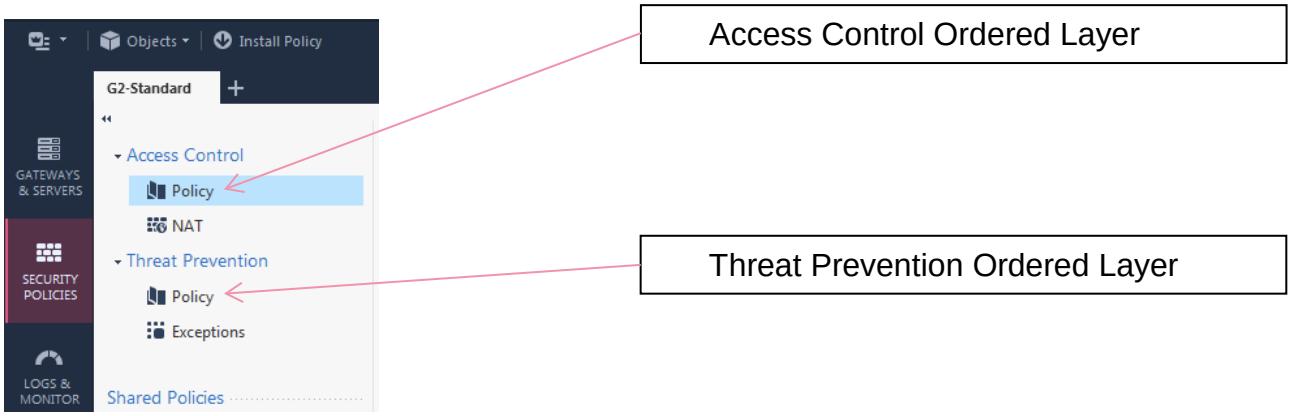
Segmentation stipulates a methodology for allowing access to other network areas, but under specific controls, restrictions, and subject to monitoring of that access.

1.2.3 Check Point Software Technologies R80.10+ Unified Policy

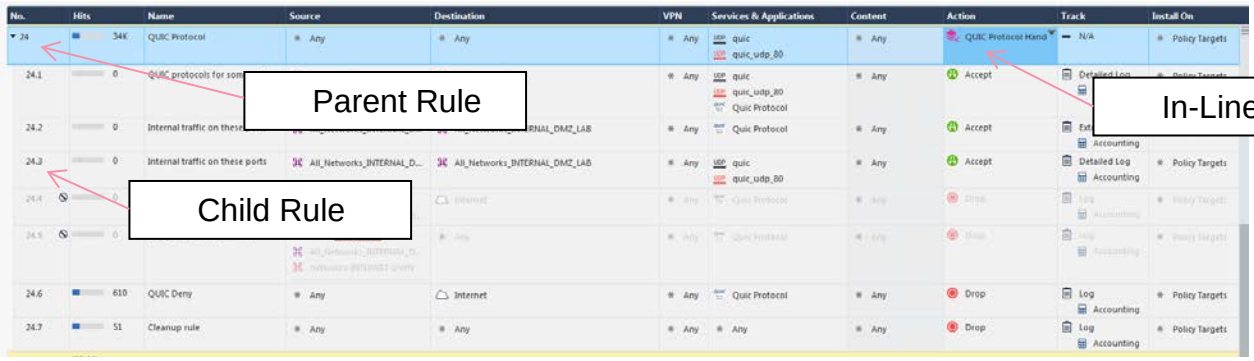
With the release of version R80.10 Check Point Software Technologies provides an updated and highly flexible policy model called Unified Policy, which consolidates the ability for Access Control policy to handle more than just a single function (firewall, application control/URL filtering, content awareness, or mobile access). Unlike versions prior to R80.10, which utilize a dedicated rule base for each function, Unified Policy can consolidate use of these (one, some, or all) together in a rule base.

With R80.10 the concept of a policy layer is introduced. A policy layer is a collection of rules, and an activation of security features (blades) for that rule base, like firewall, application control/URL filtering, content awareness, or mobile access.

Policy layers come in two (2) flavors, Ordered Layers, that are at the root of the Access Control or Threat Prevention policy sections, and execute in order, top to bottom, on a match accept for rule in an Ordered Layer, such that the next Ordered Layer is executed, until there is either a match for drop/reject or all Ordered Layers have accepted.



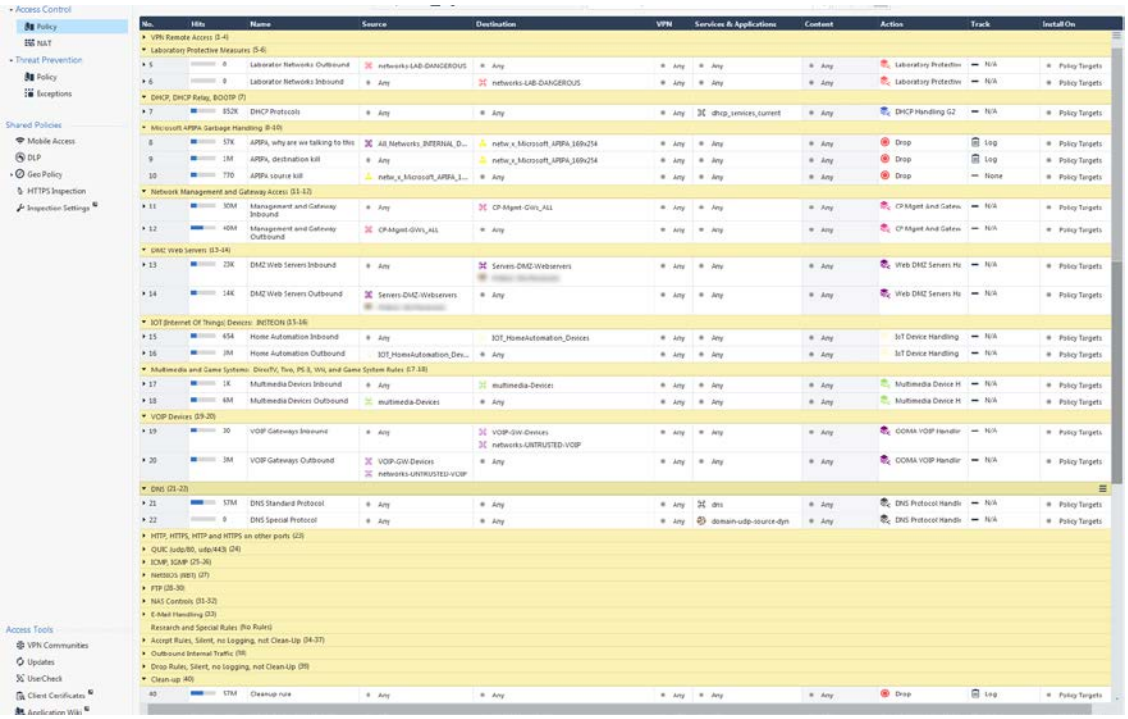
The other layer type is an Inline Layer, which means a layer is defined as the action for a parent rule, such that the Inline Layer defines the child rules of that parent rule, which may also themselves have Inline Layer actions and thus their own children. If a rule matches that has an Inline Layer as the action, matching drops into processing the rules of the Inline Layer and will either match a drop and terminate processing that packet or accept and then process any subsequent Ordered Layer(s). At no time does matching return to the level of the parent rule from an Inline Layer call.



Layers can also be set for re-use (sharing), such that the same layer can be called from different rule actions as an Inline Layer or used as Ordered Layers in the root of a policy.

The utilization of Inline Layers can be viewed like a go-to procedure/function call that does not return to the caller. Inline Layers allow segmentation of policy for specific rule approaches that can leverage efficiency, since the parent rules children are only executed if the parent matches, thus making it possible to skip rules that pertain to specific sources, destinations, services, or combinations thereof.

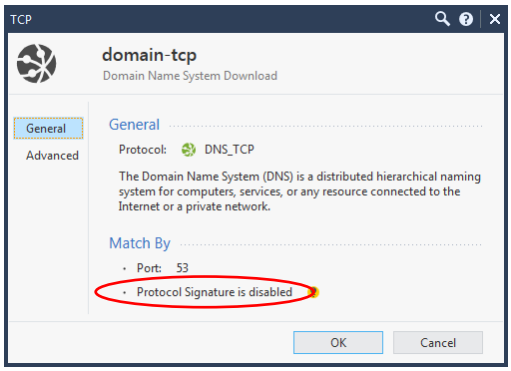
The original Standard policy in use for MyBasementCloud environment was 224 firewall rules and 26 Application Control/URL Filtering rules. By utilizing Inline Layers and the policy approach, the current MyBasementCloud G2-Standard policy in use has only 40 rules (including the clean-up and some disabled ones) in the main Ordered Layer, but most of the rules are utilizing Inline Layer calls. Notice that to ensure control, a dedicated inbound (target as destination) and outbound (target as source) rule is used with Inline Layer calls.

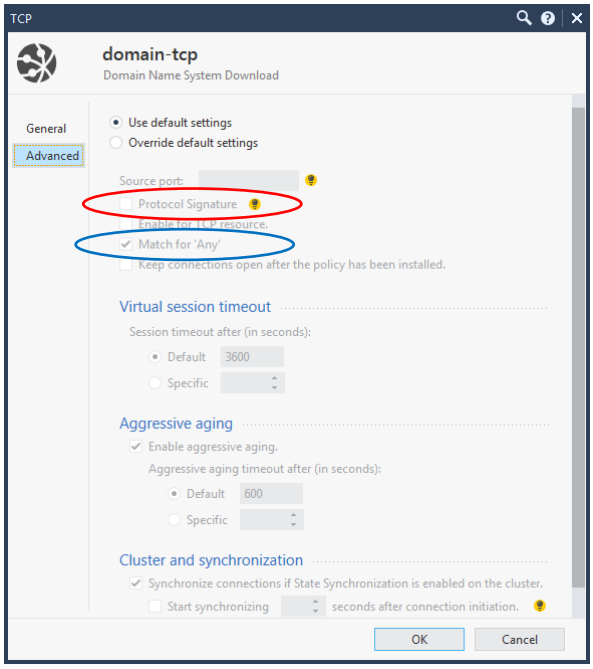


1.2.4 Check Point Software Technologies R80.10+ Protocol Signature

Protocol Signature in R80.10 is a change and overhaul of how R77.30 and prior versions handled network protocols in Application Control, to identify and ensure traffic was compliant to the specific services protocol. In R80.10 the services and their respective protocols were reviewed and utilization of a network-protocol application object was replaced with the configuration of protocol signature on certain service objects.

Protocol Signature is not enabled by default, and is an advanced property of the service object.



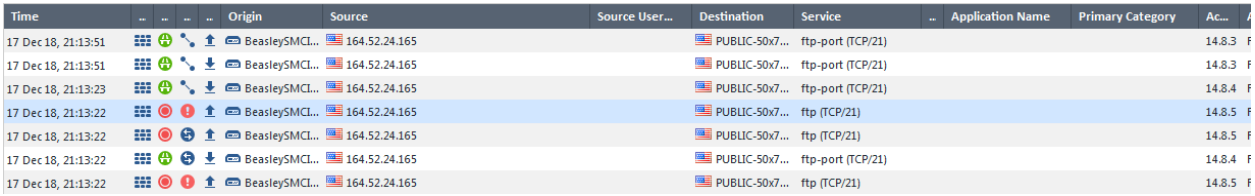


Not all services support protocol signature.

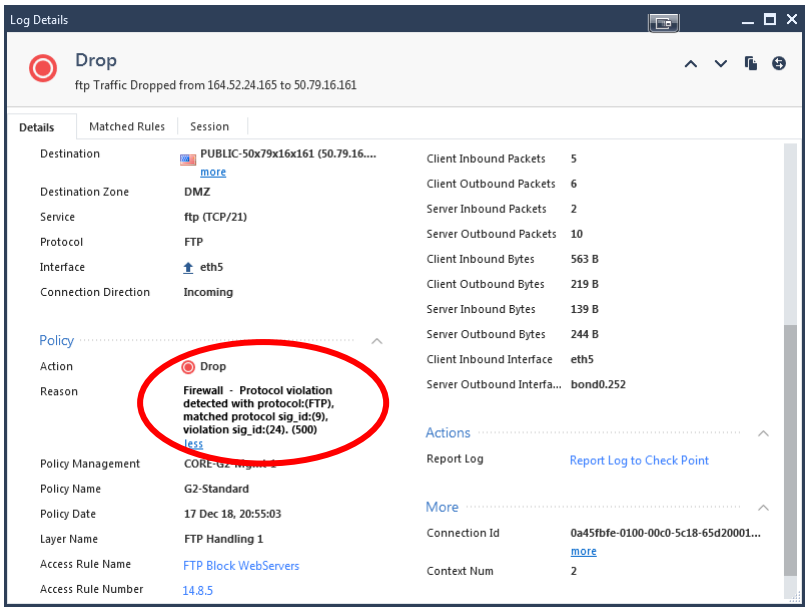
Protocol Signature also requires that the gateway and the layer where the service object is utilized has Application Control, Application Control URL Filtering (respectively) enabled to function, and will process a deeper inspection of the connection traffic to ensure RFC compliance for the services protocol.

Should traffic pass where protocol signature is expected, the access control engine will throw an alert to the logs and drop the connection due to a protocol violation.

In the example below, traffic from 164.52.24.165 to the web/ftp server starts establishing a connection via ftp on tcp port 21; however, due to a failure to adhere to protocol signature for ftp service the connection and session are dropped.



Time	Origin	Source	Source User...	Destination	Service	Application Name	Primary Category	Ac...
17 Dec 18, 21:13:51	BeasleySMCI...	164.52.24.165		PUBLIC-50x7...	ftp-port (TCP/21)			14.8.3
17 Dec 18, 21:13:51	BeasleySMCI...	164.52.24.165		PUBLIC-50x7...	ftp-port (TCP/21)			14.8.3
17 Dec 18, 21:13:23	BeasleySMCI...	164.52.24.165		PUBLIC-50x7...	ftp-port (TCP/21)			14.8.4
17 Dec 18, 21:13:22	BeasleySMCI...	164.52.24.165		PUBLIC-50x7...	ftp (TCP/21)			14.8.5
17 Dec 18, 21:13:22	BeasleySMCI...	164.52.24.165		PUBLIC-50x7...	ftp (TCP/21)			14.8.5
17 Dec 18, 21:13:22	BeasleySMCI...	164.52.24.165		PUBLIC-50x7...	ftp-port (TCP/21)			14.8.4
17 Dec 18, 21:13:22	BeasleySMCI...	164.52.24.165		PUBLIC-50x7...	ftp (TCP/21)			14.8.5



An interesting side effect of using Protocol Signature is that since connections are dropped quickly on determination of protocol failure, further Threat Prevention analysis, specifically by IPS is avoided, reducing the performance impact.

Protocol Signature does have a drawback when connecting to the unknown of the Internet, since internal to external connections can not necessarily assume that the destination operates with 100% protocol compliance, or that the internal caller is 100% protocol compliant.

2 IoT (Internet of Things) Security Challenges

IoT technologies are vulnerable to potential exploit if not protected, but also require easy access to Internet services to function and obtain configuration.

IoT devices generally have a very limited and simplified user interface for user/customer configuration, where the manufacturer stipulates that this configuration capability is sufficient for utilization according to expected operation; however, this often does not include good or deep security specific configurations.

Much of IoT communication security is handled by just using HTTPS for connections, or as is the case with OOMA VOIP phone systems, OpenVPN is utilized instead of SIP or VOIP protocols, which keeps the connection simple of HTTPS, which is often not blocked by security gateways.

Another aspect of most IoT devices is the lack of an interface for configuring network address, instead relying on accessing visible WLAN SSIDs, such that use of hidden SSID WLAN networks is not possible or more difficult, with a heavy use of pure DHCP configuration. Much of the network configuration relies on how the manufacturer provides mobile App support for configuration and access. This may stipulate a requirement for visible WLAN SSIDs, plus there is a possibility that only certain WLAN frequencies are supported, limiting use of WLAN AP technologies, which can impact segmentation approach.

2.1 HTTPS Inspection

Since many IoT and other consumer focused devices rely on HTTPS for communication with their service, even for command and control relay; the lacking configuration access for security detail configuration makes HTTPS Inspection very difficult, since certificates are all internal to the device, thus not configurable, which thwarts HTTPS Inspection, unless the HTTPS Inspection technology provider has access to the “pinned” certificates and thus can handle such inspection issues. Check Point Software Technologies is not currently able to utilize “pinned” certificates in such a scenario, as of R80.20 GA Take 101.

To ensure operation of IoT and related consumer devices, it is sadly necessary to bypass HTTPS Inspection, so that they function and can do what they were bought to do.

2.2 Product specific services and ports

Many IoT solutions focus on use of HTTPS for their primary communication with their cloud based command-and-control solutions and application access for mobile Apps, but in some cases there are other protocols utilized, like NTP, DNS, ICMP, and DHCP, but also specific port based services that are exclusive or very specific to the IoT solution.

2.2.1 Identification of Product Specific Ports and Services

Given the consumer focus of many IoT solutions, the manufacturers are not always at a level of maturity that their operative network requirements are clearly documented or publicly available, since they are assumed to be plug-and-play in consumer home networks, with minimal security interference and free access to the Internet through a consumer Internet gateway or router or WLAN router. This means that identification of

required services may require a period of observation and monitoring, or a block-and-see-what-breaks approach with reliance on good security logging.

MyBasementCloud approach is a combination of monitoring and restriction of access with logging, especially for new technologies implemented, and leverages the extensive logging capabilities of Check Point Software Technologies gateway and management solutions.

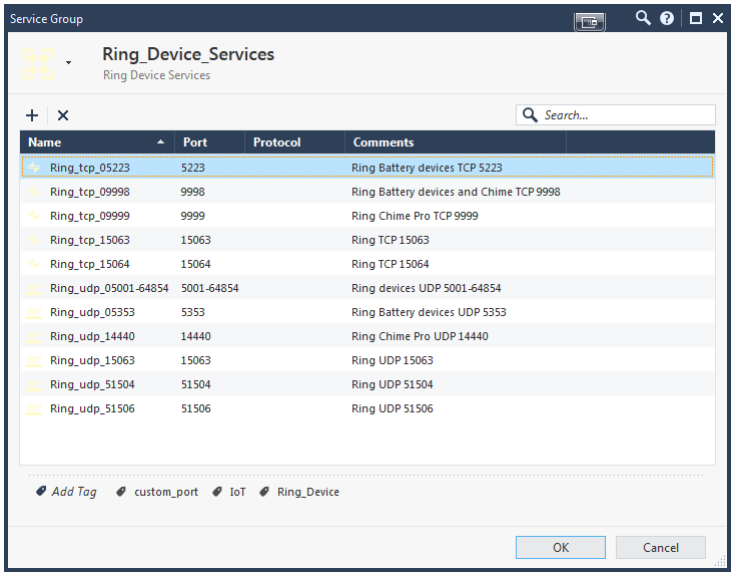
2.2.2 Some Product Specific Ports and Services

2.2.2.1 Ring

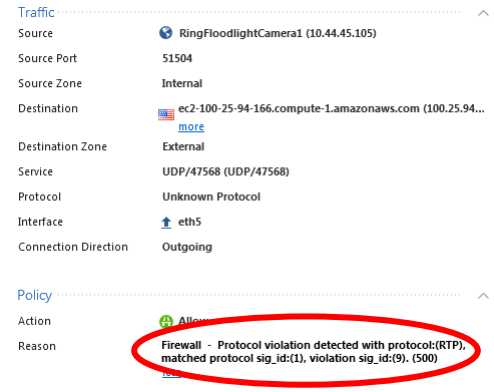
Ring devices, specifically Ring cameras and doorbell systems (security system may have additional requirements, but is not in use in MyBasementCloud infrastructure), have a number of specific ports utilized beyond the basic ones identified above. A key service is RTP, which is apparently used for audio and video transfer to cloud storage solutions. For MyBasementCloud infrastructure security configuration, a dedicated group was created to collect the port based services defined for Ring devices. This list of services is based on information provided by Ring.

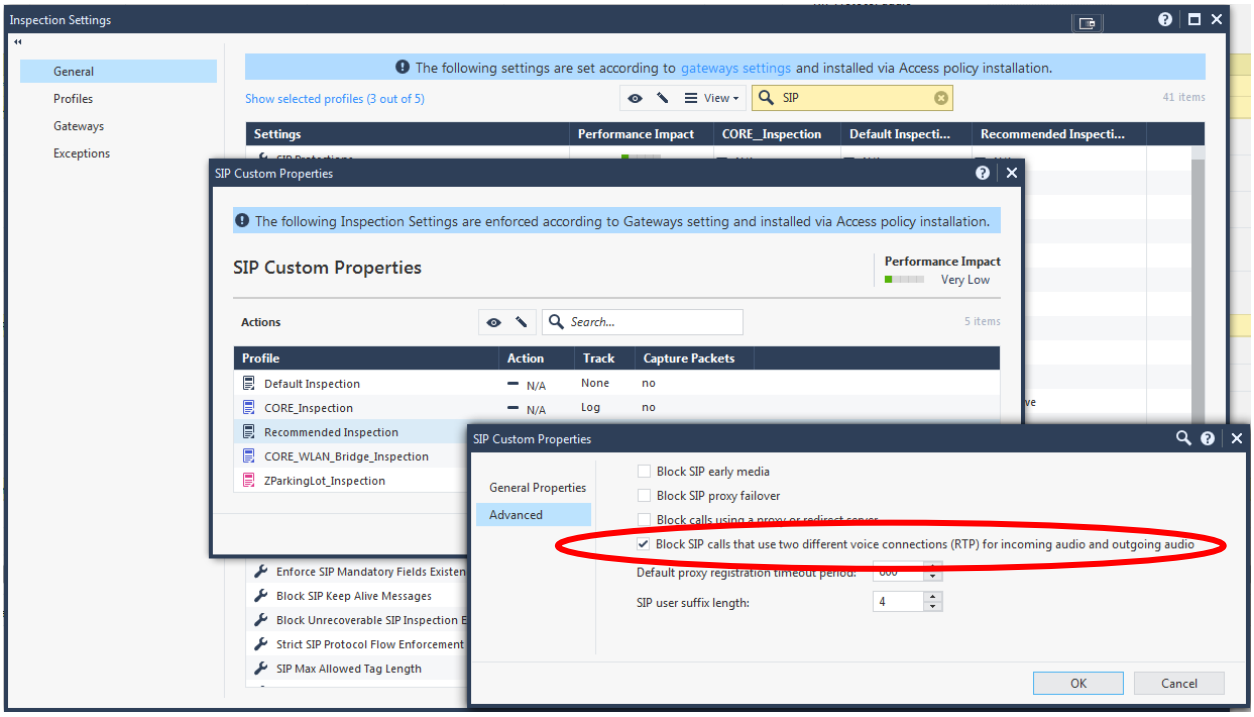
Ring ports listing:
https://support.ring.com/hc/en-us/articles/205385394-What-Ports-Do-I-Need-to-Open-in-My-Firewall-for-Ring-Doorbells-and-Chimes-

It is recommended to regularly check these Ring ports listing, since software and information updates are made and the sum of ports have changed since initial release.

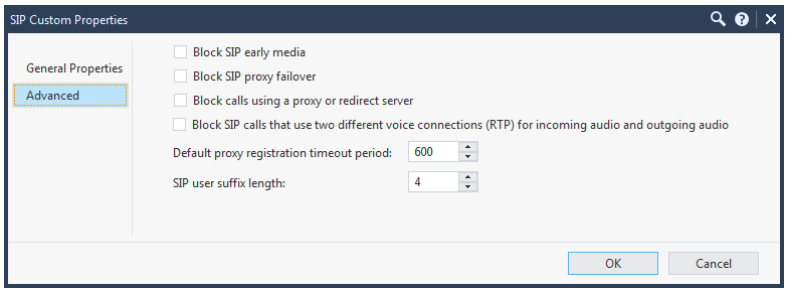


Also note the use of SIP which is utilizing RTP audio and video protocol, opening 3 streams, 2 audio and 1 video, which can lead to issues with Inspection Settings based protocol violation logs and problems with connectivity for the Ring devices. For some details review this Ring ports discussion on reddit:
https://www.reddit.com/r/ringdoorbell/comments/7ybslv/ring_devices_using_ports_outside_of_their/





MyBasementCloud has had these protocol based drops and currently implements specific SIP Custom Properties settings:

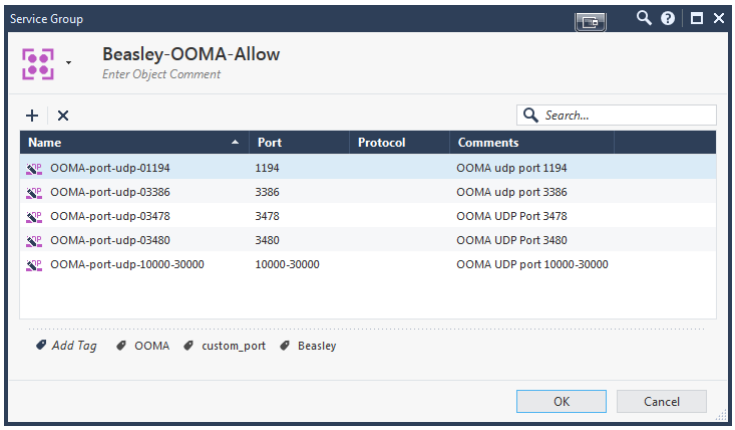


2.2.2.2 OOMA VOIP telephony solutions

OOMA is a VOIP telephony solution, replacing traditional land-line phone services with a cloud based telephony service, at significant cost savings, with a minimal monthly cost, and low initial cost for hardware.

OOMA provides a nice page of information regarding utilized ports and services, which can be found here: [Advanced Connections and Service Ports]

Key services utilized are: DNS, NTP, POP3, Syslog, and OOMA specific ports.
Key application utilized is OpenVPN, which is in lieu of VOIP technologies like SIP.



2.2.2.3 WEMO home control solutions

WEMO devices also have a set of defined services that can be explicitly defined in objects and policy. These ports can be found on the Belkin support site: https://www.belkin.com/us/support-article?articleNum=54237#_12._What_are

2.2.2.4 Ecobee thermostat/HVAC control

The Ecobee thermostat/HVAC control solution is like many IoT devices, utilizing mostly HTTPS and HTTP traffic for operations, outside of TCP port 8089, as per the information from ecobee: https://support.ecobee.com/hc/en-us/articles/227873287-Connecting-to-the-Internet.

2.2.3 Tunneling over HTTPS and QUIC

One problem with security for current IoT and consumer solutions is the utilization of UDP tunneling of information across QUIC protocol (UPD port 443) as a way to subvert security analysis, instead of using HTTPS. This also includes QUIC for HTTP over UPD port 80, again to limit security review. Blocking QUIC forces return to utilization of HTTPS.

MyBasementCloud actively thwarts utilization of QUIC protocol by actively blocking QUIC on QUIC protocol object, as well as UDP port 80 and UPD port 443.

Also, the Chrome Browser actively implements QUIC protocol as an experiment, but activates this by default.

For more information, review the following Check Point Software Technologies Secure Knowledge (SK) articles:

SK111754 - HTTP HTTPS traffic to Google services from Chrome cannot be inspected by HTTPS inspection rules
SK112249 - Best Practices - Application Control

3 Securing IoT (Internet of Things) and related Technologies

3.1 Segmentation – Network Segmentation to secure IoT

IoT security will clearly benefit from segmentation, to ensure that any potential compromise is limited, and restricted to only the exposed/exploited technology. In an ideal scenario, each technology should be separated, but this is a utopian assumption due to the realities of implementation limitation and cost—it is not viable to assign specific networks for specific vendor technologies in most environments.

3.1.1 Network Segmentation

It is highly advisable to segment and separate IoT device network traffic, both wired and wireless, from other networked systems and especially critical infrastructure. Utilization of network technologies like VLAN or dedicated network ports is a start, but should be augmented with dedicated network port (VLAN or explicit) control through a security gateway, such that entry/egress (North-South) and internal (East-West) traffic is controlled, restricted, and monitored (logging). This should ensure that compromise of an IoT device does not further compromise the rest of the environment.

3.1.2 Device Identification and control via DHCP Reservation (Known Devices)

An important issue is device identification, especially relevant when the clients are using DHCP, to ensure that assigned IP addresses are clearly identifiable to the target device. If the DHCP implementation supports reservations, it is imperative to identify the devices before deployment, to assign a known target address reservation to the device's MAC address. Solutions that provide DHCP may or may not provide an easy way to list the MAC addresses of DHCP clients that have requested and received an IP address, so work may be required to handle this MAC identification for reservation configuration in a more manual process. Good IoT devices should have their MAC addresses identified either on the device or in the packaging. For devices with an actual user interface (e.g. ecobee 3 thermostat), the MAC address should be available in the diagnostics or network configuration settings.

MyBasementCloud has implemented an extensive Microsoft Active Directory with multiple Domain Controllers running AD, DNS, DHCP, and WINS services, with two (2) Domain Controllers currently configured as DHCP servers for DHCP relay from the security gateway, operating as a primary and secondary servicing approach. This allows harvesting of identified MAC addresses not otherwise visible, from the default

DHCP requests documented in the DHCP administrative management console (MMC), such that either a direct reservation can be created or the information used to manually create such a reservation. This definition of a DHCP reservation for specific devices allows creation of device specific host objects in Check Point Software Technologies management environment.

By clearly identifying the IoT devices with explicit DHCP reservations (for those that can't be configured with fixed addresses), it is possible to utilize the associated host objects and create device manufacturer or device type specific groups of hosts for easy utilization security policy.

Combinations of Microsoft scripting for DHPC configuration and Check Point Security Technologies Management API (in R80 and higher management environments) allow for quick configuration of multiple devices.

3.1.3 Segmentation versus Isolation

An alternative to segmentation with utilization of shared infrastructure and good security policy for security gateways is to fall back to complete isolation, such that IoT devices connect on a dedicated network environment that does not have network connectivity or systems (gateways, routers, security gateways) shared with other environments. This may not be plausible in all instances, and as such is not the solution addressed here.

While possible to execute in the MyBasementCloud infrastructure, an Isolation approach would require utilization of dedicated networking equipment and not provide a clearly controlled approach with observable network traffic (logged on security gateway) or actual easy visibility of what IoT devices are active, working [or not], and how they communicate.

3.1.4 MyBasementCloud best practices

MyBasementCloud executes the network segmentation via dedicated VLANs, and where needed, interfaces to explicitly separate traffic for the following:

- IoT devices network
- VOIP device network (includes alarm system)
- Multimedia and game system device network (no shared media access required)

These specific device networks are controlled in how they are able to interact with other networks, how they access the Internet, and how they can be accessed from the Internet. In most cases, direct Internet access is blocked, so only return traffic on a connection is allowed, which is sufficient for most IoT operation.

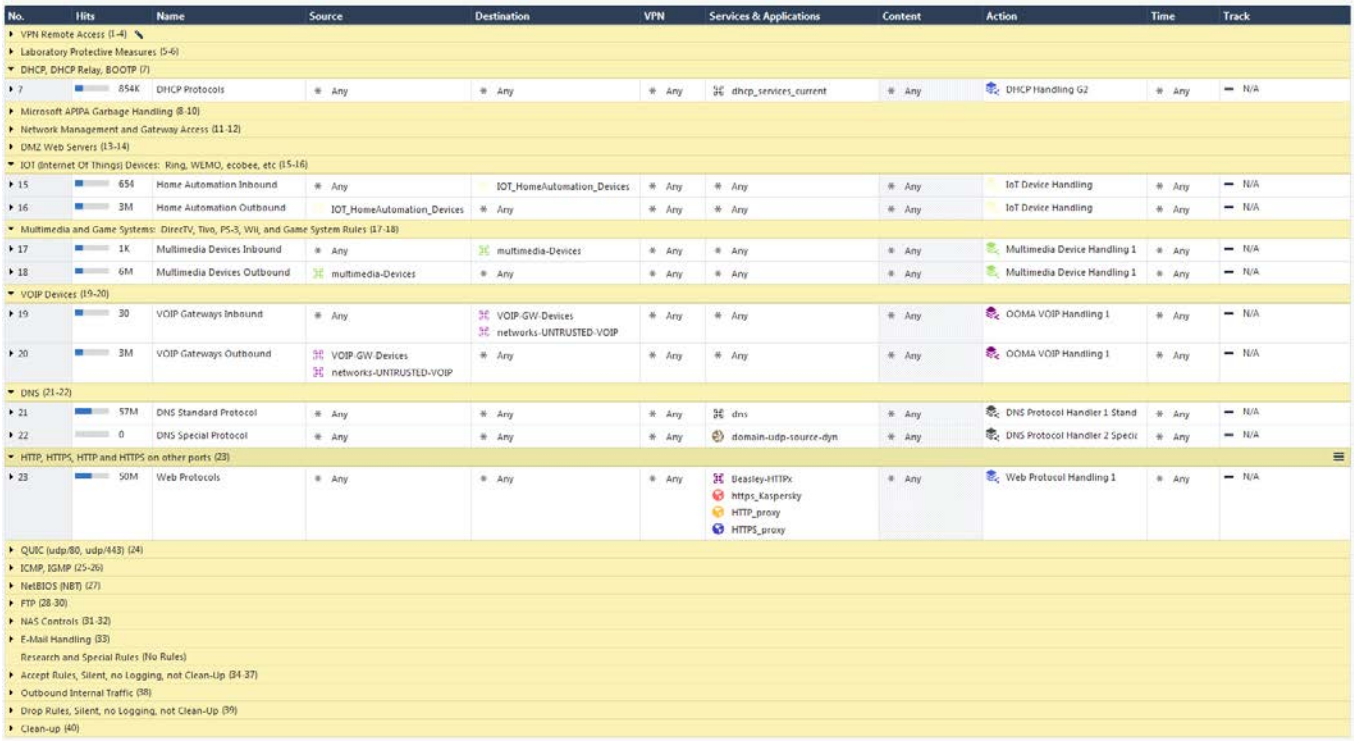
Due to the use of shared services for DHCP, security gateway implementation of DHCP relay allows for central administration of addressing and DCHP services to the IoT and related device networks. Appropriate controls via the Access Control rule base are required to ensure proper DHCP function in the segmented networks.

3.2 Unified Policy and utilization of Inline Layers

Network segmentation is a good first step, but it is also beneficial to look at how to approach security policy segmentation, to both leverage easier definition of this policy, and also take advantage of a performance enhancing approach.

R80.10 and later Unified Policy provides Inline Layers to help easily segment a policy rule base, and leverage the improved rule matching approach to reduce processing unnecessary policy rules.

MyBasementCloud infrastructure actively leverages numerous shared Inline Layers to achieve easy segmentation for specific purposes (hosts, services, etc.), and then utilizes those Inline Layers further to simplify, for a view of the top level of the main Access Control Ordered Layer, see below.



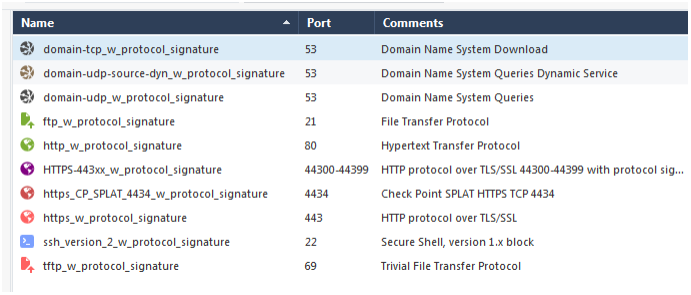
No.	Hits	Name	Source	Destination	VPN	Services & Applications	Content	Action	Time	Track
VPN Remote Access (1-4)										
Laboratory Protective Measures (5-6)										
DHCP, DHCP Relay, BOOTP (7)										
7	854K	DHCP Protocols	Any	Any	Any	dhcp_services_current	Any	DHCP Handling G2	Any	N/A
Microsoft APIPA Garbage Handling (8-10)										
Network Management and Gateway Access (11-12)										
DMZ Web Servers (13-14)										
IOT (Internet Of Things) Devices: Ring, WEMO, ecobee, etc (15-16)										
15	654	Home Automation Inbound	Any	IOT_HomeAutomation_Devices	Any	Any	Any	IoT Device Handling	Any	N/A
16	3M	Home Automation Outbound	IOT_HomeAutomation_Devices	Any	Any	Any	Any	IoT Device Handling	Any	N/A
Multimedia and Game Systems: DirecTV, Tivo, PS-3, Wii, and Game System Rules (17-18)										
17	1K	Multimedia Devices Inbound	Any	multimedia-Devices	Any	Any	Any	Multimedia Device Handling 1	Any	N/A
18	6M	Multimedia Devices Outbound	multimedia-Devices	Any	Any	Any	Any	Multimedia Device Handling 1	Any	N/A
VOIP Devices (19-20)										
19	30	VOIP Gateways Inbound	Any	VOIP-GW-Devices networks-UNTRUSTED-VOIP	Any	Any	Any	OOMA VOIP Handling 1	Any	N/A
20	3M	VOIP Gateways Outbound	VOIP-GW-Devices networks-UNTRUSTED-VOIP	Any	Any	Any	Any	OOMA VOIP Handling 1	Any	N/A
DNS (21-22)										
21	57M	DNS Standard Protocol	Any	Any	Any	dns	Any	DNS Protocol Handler 1 Stand	Any	N/A
22	0	DNS Special Protocol	Any	Any	Any	domain-udp-source-dyn	Any	DNS Protocol Handler 2 Speci	Any	N/A
HTTP, HTTPS, HTTP and HTTPS on other ports (23)										
23	50M	Web Protocols	Any	Any	Any	Beasley-HTTPx https_Kaspersky HTTP_proxy HTTPS_proxy	Any	Web Protocol Handling 1	Any	N/A
QUIC (udp/80, udp/443) (24)										
ICMP, IGMP (25-26)										
NetBIOS (NBT) (27)										
FTP (28-30)										
NAS Controls (31-32)										
E-Mail Handling (33)										
Research and Special Rules (No Rules)										
Accept Rules, Silent, no Logging, not Clean-Up (34-37)										
Outbound Internal Traffic (38)										
Drop Rules, Silent, no Logging, not Clean-Up (39)										
Clean-up (40)										

3.3 Protocol Signature – Enforcing RFC compliant operation of services utilized by IoT

For certain IoT services, especially in segmented environments (versus isolated ones), the use of Protocol Signature can assure that communicate from an IoT device to internal assets (or external assets if necessary) are compliant with the protocols RFC implementation requirements, and that a breach of this compliance leads to a termination of the connection(s) and session.

3.3.1 Protocol Signature Best Practice recommendations

CHECK POINT BEST PRACTICE RECOMMENDATION: When enabling Protocol Signature, it is advisable to CLONE the service object and name the clone appropriately, e.g. domain-tcp_w_protocol_signature. Adding a Tag to the service identify that Protocol Signature is enabled, also helps quickly finding the service object in the SmartConsole Object Explorer.



Name	Port	Comments
domain-tcp_w_protocol_signature	53	Domain Name System Download
domain-udp-source-dyn_w_protocol_signature	53	Domain Name System Queries Dynamic Service
domain-udp_w_protocol_signature	53	Domain Name System Queries
ftp_w_protocol_signature	21	File Transfer Protocol
http_w_protocol_signature	80	Hypertext Transfer Protocol
HTTPS-443xx_w_protocol_signature	44300-44399	HTTP protocol over TLS/SSL 44300-44399 with protocol sig...
https_CP_SPLAT_4434_w_protocol_signature	4434	Check Point SPLAT HTTPS TCP 4434
https_w_protocol_signature	443	HTTP protocol over TLS/SSL
ssh_version_2_w_protocol_signature	22	Secure Shell, version 1.x block
tftp_w_protocol_signature	69	Trivial File Transfer Protocol

MyBasementCloud BEST PRACTICE RECOMMENDATION: On the clone object enabling Protocol Signature, uncheck the “Match for Any” option to ensure absolute granular control of the clone service object utilization and matching.

MyBasementCloud utilizes protocol signature on specific services (e.g. FTP, HTTP, HTTPS) on inbound connections to DMZ hosts, and for DNS on internal hosts to internal DNS servers, but connections to external hosts and services are not subject to protocol signature to ensure connectivity. Microsoft is notorious for not being RFC protocol compliant, and after MyBasementCloud tests, internal to external use of protocol signature discontinued.

3.3.2 Protocol Signature for critical network services – DNS

The utilization of tunneling over DNS in exploit scenarios, leads to a need to implement appropriate Protocol Signature control for DNS services, especially for accessing internal DNS hosts.

No.	Hits	Name	Source	Destination	VPN	Services & Applications	Content	Action	Time	Track
16	3M	Home Automation Outbound	IOT_HomeAutomation_Devices	Any	Any	Any	Any	IoT Device Handling	Any	N/A
IoT Common Protocols (16.1-16.5)										
16.1	0	DHCP and DHCP Relay Operations	Any	Any	Any	dhcp_services_current	Any	DHCP Handling G2	Any	N/A
16.2	892K	IoT Outbound DNS	IOT_HomeAutomation_Devices	Any	Any	dns	Any	DNS Protocol Handler 1 Stand	Any	N/A
Internal DNS Server Queries (16.2.1-16.2.5)										
Check Point Infra DNS Queries (16.2.6-16.2.8)										
Multimedia and IOT DNS Queries (16.2.9-16.2.11)										
16.2.9	1M	IOS and Multimedia DNS Query internal	multimedia-Devices, IOS_Mobile_Devices, IOT_HomeAutomation_Devices	Servers-DNS-Internal	Any	dns_w_protocol_signature	Any	Accept	Any	None
16.2.10	354K	IOS and Multimedia DNS Query	multimedia-Devices, IOS_Mobile_Devices	Any	Any	dns	Any	Accept	Any	Log
16.2.11	248K		IOT_HomeAutomation_Devices	Any	Any	dns	Any	Accept	Any	Log
Infrastructure DNS Queries (16.2.12-16.2.13)										
Query to Internal DNS (16.2.14-16.2.17)										
Query External DNS (16.2.18-16.2.19)										
domain-udp source dyn (16.2.20-16.2.21)										
Internal DNS Queries, not protocol signature (16.2.22-16.2.23)										
Clean-up Unexpected DNS (16.2.24)										
16.2.24	529	Cleanup rule	Any	Any	Any	Any	Any	Drop	Any	Log

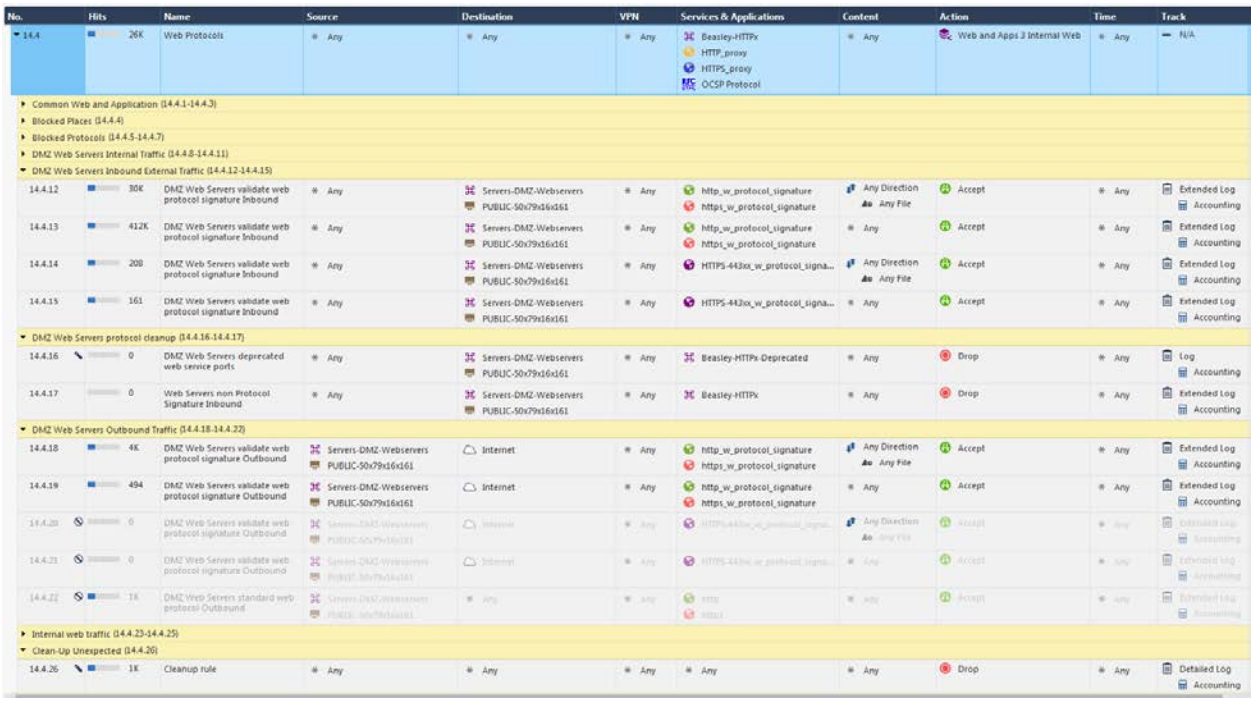
The example policy for IoT outbound/inbound traffic in MyBasementCloud above, utilizes a shared Inline Layer approach, with a specific rule entry for handling DNS, again via a dedicated shared Inline Layer call for a DNS handling layer. This DNS layer

includes some very detailed (the engineer's CDO) logging and controls for DNS handling, but has explicit controls for IoT and related device's DNS queries, such that queries to internal DNS are subject to protocol signature with the dns_w_protocol_signature service group object being used, while external queries are handled with the default DNS service group object. This ensures that internal queries are safe, even if the IoT device is compromised and the attacker attempts to use exploits over DNS tunneling (which will fail the protocol test and get terminated). External DNS queries are handled via non-Protocol Signature DNS service, so that operations are not impeded; however, this could be changed if protecting other DNS services was an objective.

3.3.3 Protocol Signature for general inbound traffic on vulnerable network services – HTTP/HTTPS, FTP

While not necessarily relevant for IoT security policy, which generally means outbound traffic, it is worth looking into Protocol Signature approach for inbound services to DMZ servers with host access for HTTP, HTTPS, and FTP services, as these are continually targetted from external entities, looking for vulnerabilities.

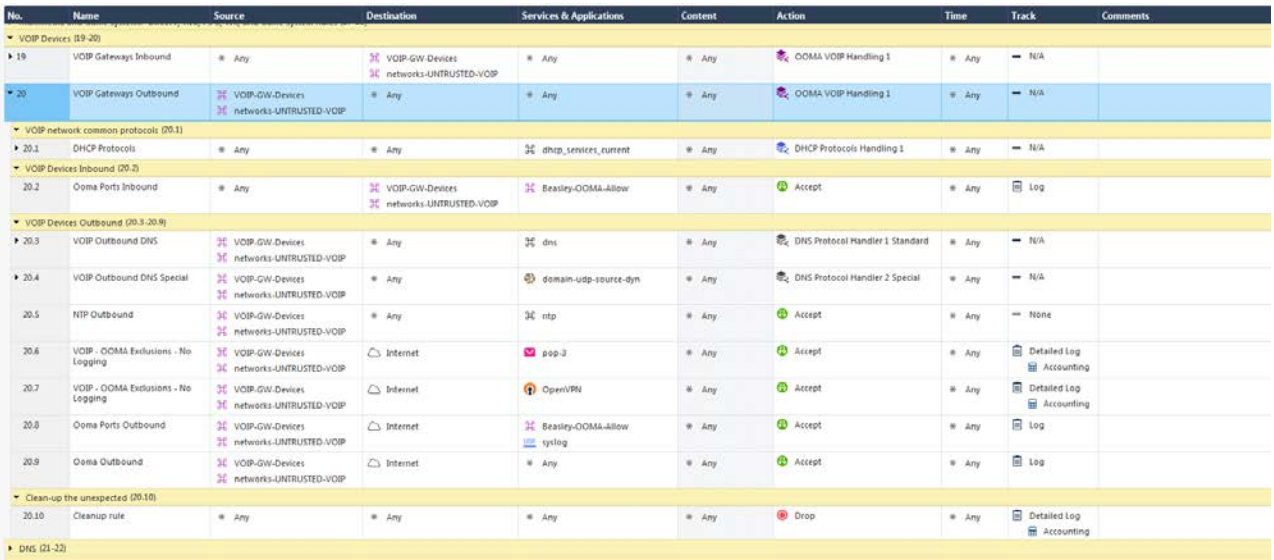
MyBasementCloud infrastructure implements protocol signature controls for HTTP and HTTPS as well as FTP for access to DMZ web servers, to ensure they are protected from external threats.



3.4 Example Access Control Policy for Securing IoT (Internet of Things) devices

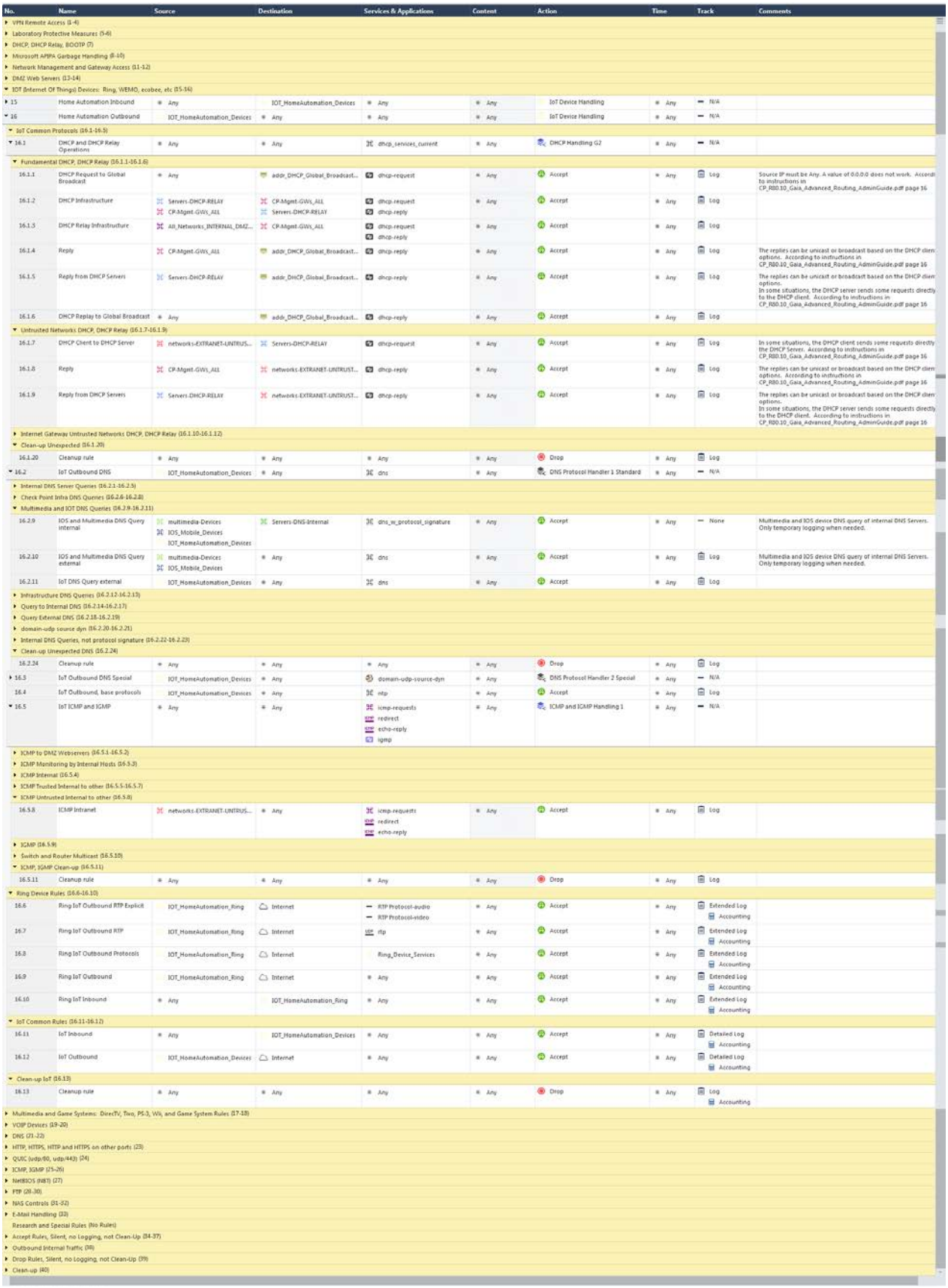
The following examples from MyBasementCloud infrastructure show the policy approach used for IoT solutions in use.

3.4.1 OOMA VOIP devices and ADT Pulse Security Policy elements



Details for the DHCP and DNS Protocol Handler Inline Layers are expanded to relevant elements in the IoT devices Policy elements section.

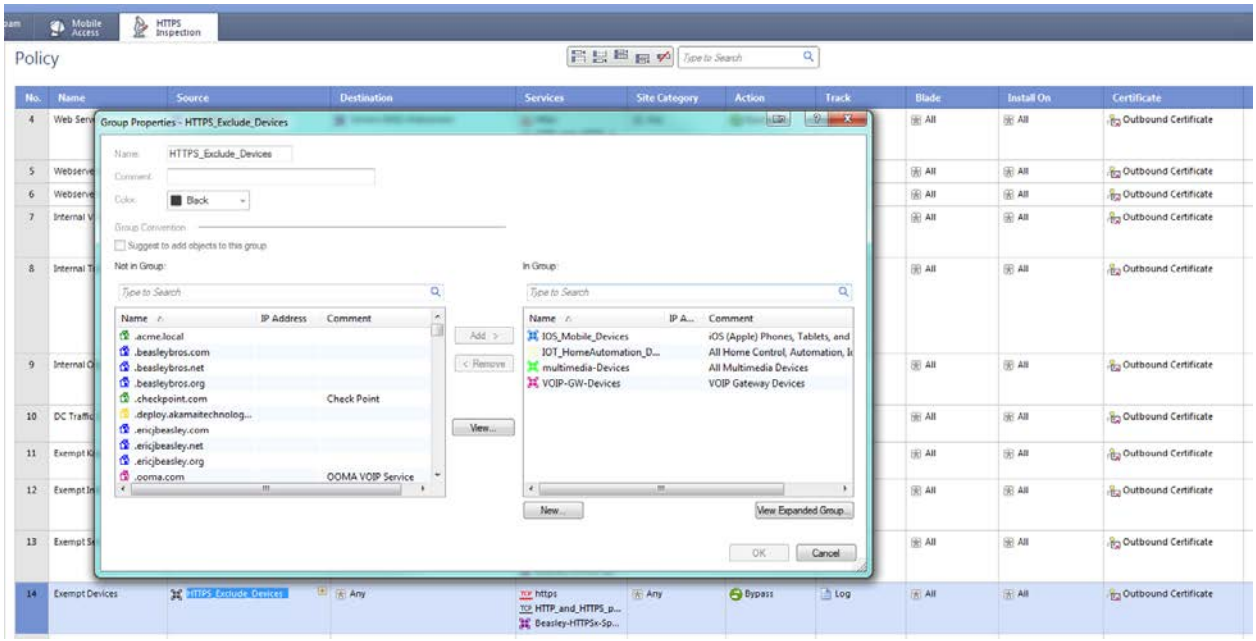
3.4.2 IoT devices Policy elements



No.	Name	Source	Destination	Services & Applications	Content	Action	Time	Track	Comments
VPN Remote Access (1-4)									
Laboratory Protective Measures (5-6)									
DHCP, DHCP Relay, BOOTP (7)									
Microsoft APIPA Garbage Handling (8-10)									
Network Management and Gateway Access (11-12)									
DMZ Web Servers (13-14)									
IOT (Internet Of Things) Devices: Ring, WEMO, ecobee, etc (15-16)									
15	Home Automation Inbound	Any	IOT_HomeAutomation_Devices	Any	Any	IoT Device Handling	Any	N/A	
16	Home Automation Outbound	IOT_HomeAutomation_Devices	Any	Any	Any	IoT Device Handling	Any	N/A	
IoT Common Protocols (16.1-16.5)									
16.1	DHCP and DHCP Relay Operations	Any	Any	dhcp_services_current	Any	DHCP Handling G2	Any	N/A	
Fundamental DHCP, DHCP Relay (16.1.1-16.1.6)									
16.1.1	DHCP Request to Global Broadcast	Any	addr_DHCP_Global_Broadcast...	dhcp-request	Any	Accept	Any	Log	Source IP must be Any. A value of 0.0.0.0 does not work. Accordi to instructions in CP_R80.10_Gaia_Advanced_Routing_AdminGuide.pdf page 16
16.1.2	DHCP Infrastructure	Servers-DHCP-RELAY CP-Mgmt-GWs_ALL	CP-Mgmt-GWs_ALL Servers-DHCP-RELAY	dhcp-request dhcp-reply	Any	Accept	Any	Log	
16.1.3	DHCP Relay Infrastructure	All_Networks_INTERNAL_DMZ...	CP-Mgmt-GWs_ALL	dhcp-request dhcp-reply	Any	Accept	Any	Log	
16.1.4	Reply	CP-Mgmt-GWs_ALL	addr_DHCP_Global_Broadcast...	dhcp-reply	Any	Accept	Any	Log	The replies can be unicast or broadcast based on the DHCP clien options. According to instructions in CP_R80.10_Gaia_Advanced_Routing_AdminGuide.pdf page 16
16.1.5	Reply from DHCP Servers	Servers-DHCP-RELAY	addr_DHCP_Global_Broadcast...	dhcp-reply	Any	Accept	Any	Log	The replies can be unicast or broadcast based on the DHCP clien options. In some situations, the DHCP server sends some requests directly to the DHCP client. According to instructions in CP_R80.10_Gaia_Advanced_Routing_AdminGuide.pdf page 16
16.1.6	DHCP Replay to Global Broadcast	Any	addr_DHCP_Global_Broadcast...	dhcp-reply	Any	Accept	Any	Log	
Untrusted Networks DHCP, DHCP Relay (16.1.7-16.1.9)									
16.1.7	DHCP Client to DHCP Server	networks-EXTRANET-UNTRUS...	Servers-DHCP-RELAY	dhcp-request	Any	Accept	Any	Log	In some situations, the DHCP client sends some requests directly the DHCP Server. According to instructions in CP_R80.10_Gaia_Advanced_Routing_AdminGuide.pdf page 16
16.1.8	Reply	CP-Mgmt-GWs_ALL	networks-EXTRANET-UNTRUST...	dhcp-reply	Any	Accept	Any	Log	The replies can be unicast or broadcast based on the DHCP clien options. According to instructions in CP_R80.10_Gaia_Advanced_Routing_AdminGuide.pdf page 16
16.1.9	Reply from DHCP Servers	Servers-DHCP-RELAY	networks-EXTRANET-UNTRUST...	dhcp-reply	Any	Accept	Any	Log	The replies can be unicast or broadcast based on the DHCP clien options. In some situations, the DHCP server sends some requests directly to the DHCP client. According to instructions in CP_R80.10_Gaia_Advanced_Routing_AdminGuide.pdf page 16
Internet Gateway Untrusted Networks DHCP, DHCP Relay (16.1.10-16.1.12)									
Clean-up Unexpected (16.1.20)									
16.1.20	Cleanup rule	Any	Any	Any	Any	Drop	Any	Log	
16.2	IoT Outbound DNS	IOT_HomeAutomation_Devices	Any	dns	Any	DNS Protocol Handler 1 Standard	Any	N/A	
Internal DNS Server Queries (16.2.1-16.2.5)									
Check Point Infra DNS Queries (16.2.6-16.2.8)									
Multimedia and IOT DNS Queries (16.2.9-16.2.11)									
16.2.9	IOS and Multimedia DNS Query internal	multimedia-Devices IOS_Mobile_Devices IOT_HomeAutomation_Devices	Servers-DNS-Internal	dns_w_protocol_signature	Any	Accept	Any	None	Multimedia and IOS device DNS query of internal DNS Servers. Only temporary logging when needed.
16.2.10	IOS and Multimedia DNS Query external	multimedia-Devices IOS_Mobile_Devices	Any	dns	Any	Accept	Any	Log	Multimedia and IOS device DNS query of internal DNS Servers. Only temporary logging when needed.
16.2.11	IoT DNS Query external	IOT_HomeAutomation_Devices	Any	dns	Any	Accept	Any	Log	
Infrastructure DNS Queries (16.2.12-16.2.13)									
Query to Internal DNS (16.2.14-16.2.17)									
Query External DNS (16.2.18-16.2.19)									
domain-udp source dyn (16.2.20-16.2.21)									
Internal DNS Queries, not protocol signature (16.2.22-16.2.23)									
Clean-up Unexpected DNS (16.2.24)									
16.2.24	Cleanup rule	Any	Any	Any	Any	Drop	Any	Log	
16.3	IoT Outbound DNS Special	IOT_HomeAutomation_Devices	Any	domain-udp-source-dyn	Any	DNS Protocol Handler 2 Special	Any	N/A	
16.4	IoT Outbound, base protocols	IOT_HomeAutomation_Devices	Any	ntp	Any	Accept	Any	Log	
16.5	IoT ICMP and IGMP	Any	Any	icmp-requests redirect echo-reply igmp	Any	ICMP and IGMP Handling 1	Any	N/A	
ICMP to DMZ Webservers (16.5.1-16.5.2)									
ICMP Monitoring by Internal Hosts (16.5.3)									
ICMP Internal (16.5.4)									
ICMP Trusted Internal to other (16.5.5-16.5.7)									
ICMP Untrusted Internal to other (16.5.8)									
16.5.8	ICMP Intranet	networks-EXTRANET-UNTRUS...	Any	icmp-requests redirect echo-reply	Any	Accept	Any	Log	
IGMP (16.5.9)									
Switch and Router Multicast (16.5.10)									
ICMP, IGMP Clean-up (16.5.11)									
16.5.11	Cleanup rule	Any	Any	Any	Any	Drop	Any	Log	
Ring Device Rules (16.6-16.10)									
16.6	Ring IoT Outbound RTP Explicit	IOT_HomeAutomation_Ring	Internet	RTP Protocol-audio RTP Protocol-video	Any	Accept	Any	Extended Log Accounting	
16.7	Ring IoT Outbound RTP	IOT_HomeAutomation_Ring	Internet	rtp	Any	Accept	Any	Extended Log Accounting	
16.8	Ring IoT Outbound Protocols	IOT_HomeAutomation_Ring	Internet	Ring_Device_Services	Any	Accept	Any	Extended Log Accounting	
16.9	Ring IoT Outbound	IOT_HomeAutomation_Ring	Internet	Any	Any	Accept	Any	Extended Log Accounting	
16.10	Ring IoT Inbound	Any	IOT_HomeAutomation_Ring	Any	Any	Accept	Any	Extended Log Accounting	
IoT Common Rules (16.11-16.12)									
16.11	IoT Inbound	Any	IOT_HomeAutomation_Devices	Any	Any	Accept	Any	Detailed Log Accounting	
16.12	IoT Outbound	IOT_HomeAutomation_Devices	Internet	Any	Any	Accept	Any	Detailed Log Accounting	
Clean-up IoT (16.13)									
16.13	Cleanup rule	Any	Any	Any	Any	Drop	Any	Log Accounting	
Multimedia and Game Systems: DirecTV, Tivo, PS-3, Wii, and Game System Rules (17-18)									
VOIP Devices (19-20)									
DNS (21-22)									
HTTP, HTTPS, HTTP and HTTPS on other ports (23)									
QUIC (udp/80, udp/443) (24)									
ICMP, IGMP (25-26)									
NetBIOS (NBT) (27)									
FTP (28-30)									
NAS Controls (31-32)									
E-Mail Handling (33)									
Research and Special Rules (No Rules)									
Accept Rules, Silent, no Logging, not Clean-Up (34-37)									
Outbound Internal Traffic (38)									
Drop Rules, Silent, no Logging, not Clean-Up (39)									
Clean-up (40)									

3.5 Related operations: HTTPS Inspection; Bypass Workaround

Given the inherent challenges with IoT device configuration and a lack of access, HTTPS Inspection poses a serious challenge, so the easiest approach to ensure function and operation is exclusion of the devices to ensure HTTPS Inspection bypass, using the recommendation for collecting IoT devices into clearly identifiable groups.



HTTPS Inspection in R80.10 and R80.20 does not allow use of applications in the services column of the HTTPS Inspection rule base, so using the application as a reference does not work, and only the network source destination is a reasonable working alternative. Potentially, with significant effort, identification of the IoT devices host services Internet domains could be used to create Application-Site objects and used in the Service column, but this is not a reasonable chore.

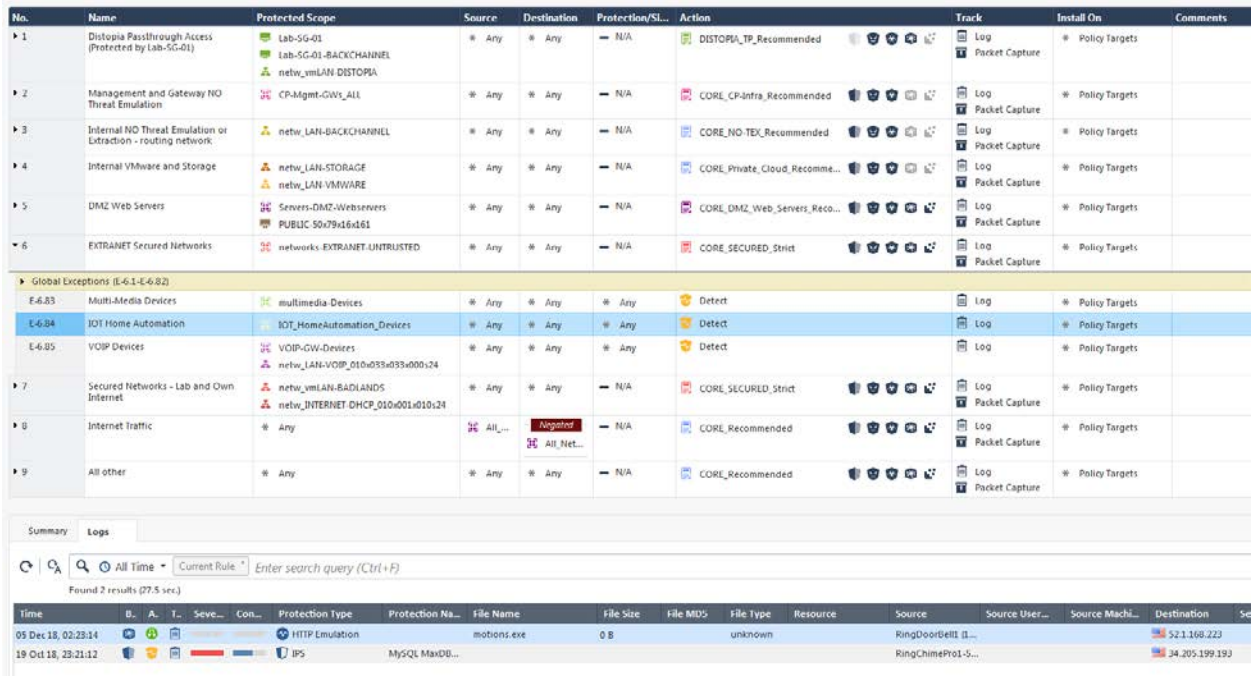
Importantly, by explicitly identifying the devices by host object and grouping those, other things that might wander onto the IoT network segments fall victim to being HTTPS Inspected by default rule (inspect all rest), thwarting easy exploitation of the segmented network by other devices or invaders.

3.6 Related operations: Threat Prevention policy

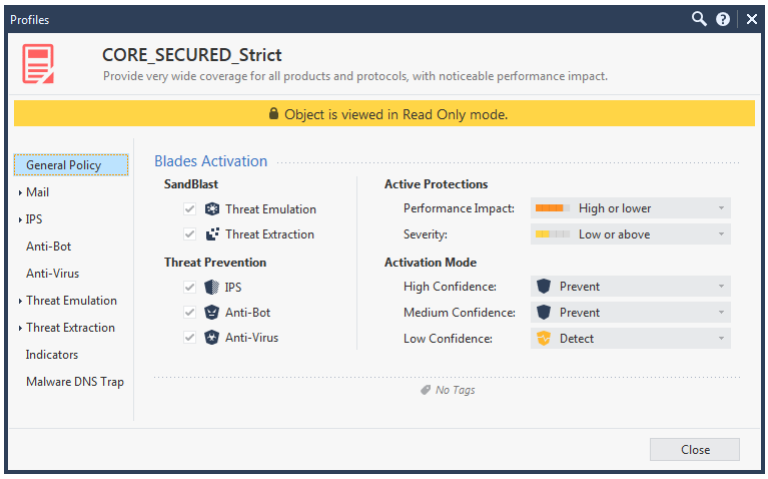
Threat Prevention policy is another aspect that can be impacting to IoT implementation, due to the “cloud” nature of these systems, so a strict and tight security approach may result in serious impact to the operation of these devices, but ignoring them is not an option either, so a monitoring approach may be the best idea, ideally again utilizing the specific of identified devices to ensure only known actors are monitored while unknown devices are specifically prevented.

MyBasementCloud currently implements focused Threat Prevention policy approach with dedicated Threat Prevention security profiles applied to enable granular control of

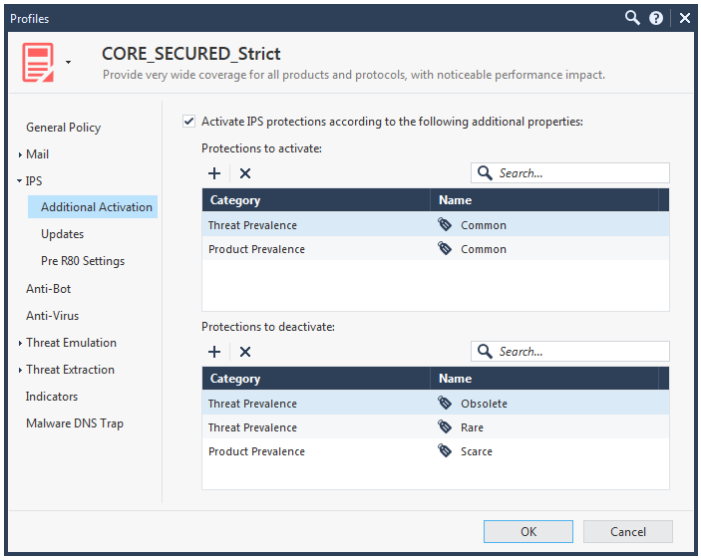
actions by the Threat Prevention technology blades. Note that explicit scope elements are defined to utilize exceptions forcing Detect versus Prevent to ensure operation, but this does not exclude unknown devices from Prevent protection. Also observe the Threat Emulation executed against the Ring Doorbell's file download of motions.exe, an apparent upgrade of the system firmware.



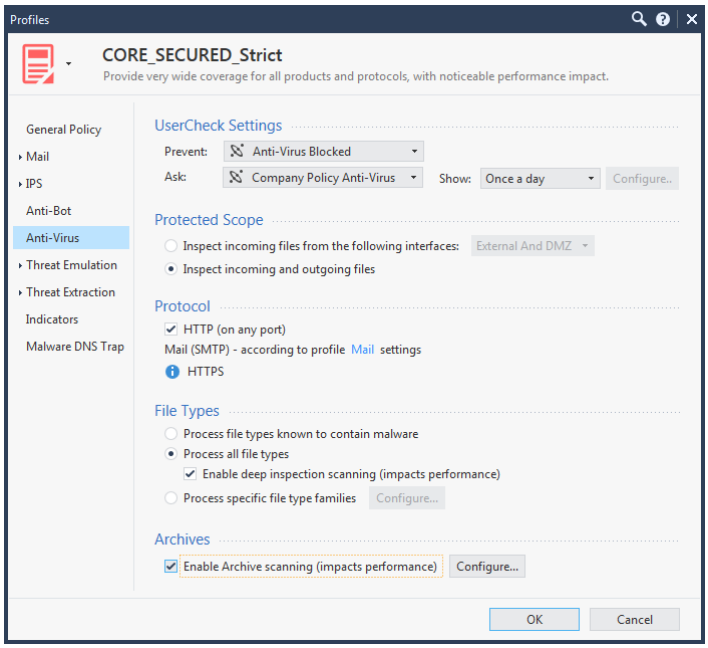
The Threat Prevention Profile used for IoT, or more generally Untrusted Extranet networks is the CORE_SECURED_Strict profile cloned from the R80.10 Strict profile, which is a strong starting point. While MyBasementCloud currently does not implement Threat Extraction due to a lack of internal e-mail servers, thus no ability to utilize the MTA for e-mail analysis, the profile has it enabled for future operational changes.



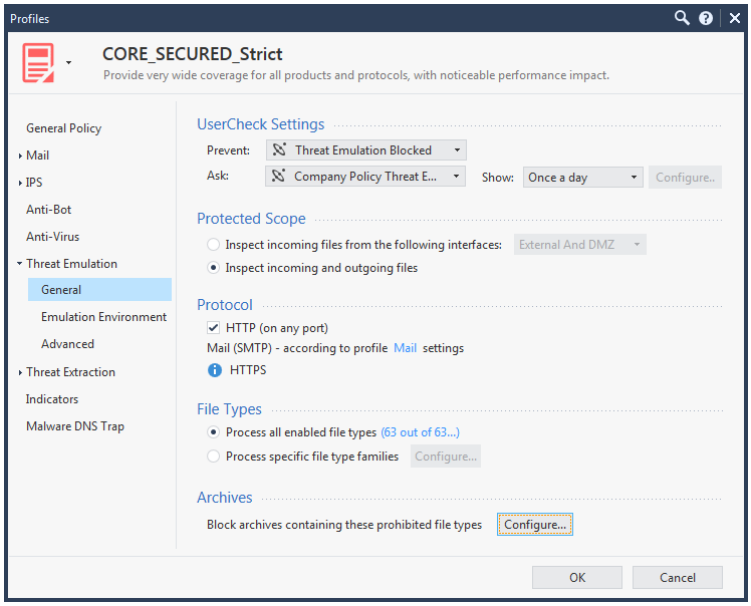
Threat Prevention profile IPS activation settings are currently open, since no clear technology has been identified that needs protection, instead Threat Prevalence and Product Prevalence is utilized for activation and deactivation of IPS protections.

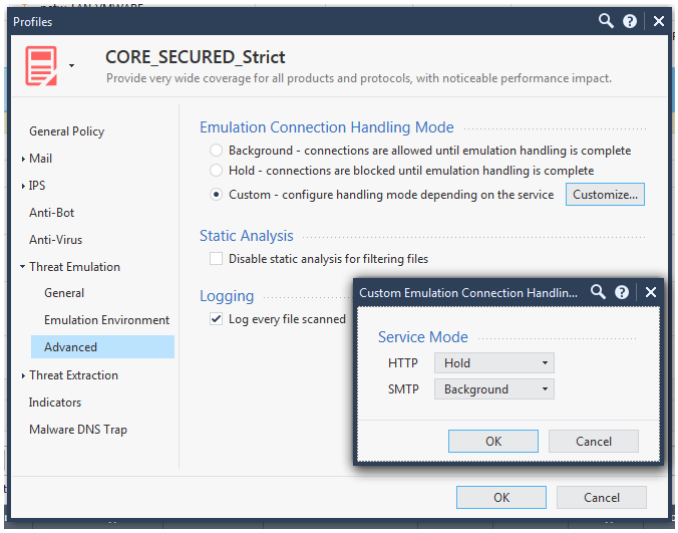


Threat Prevention profile Anti-Virus configuration focuses on both incoming and outgoing files to the scoped protected object with detail enforcement of inspection for all file types with deep inspection and archive scanning.



Threat Prevent Profile Threat Emulation General settings are also focused on a strict security approach with inspection of inbound and outbound files on all file types. Under the Advance settings, this environment, lacking an MTA handles HTTP with a hold operation, while handling SMTP in background mode. All scanned files are logged to provide information.





Threat Prevention profile configuration of Indicators [of Compromise (IOCs)] is common to all Threat Prevention profiles, so any IOCs identified are used. Malware DNS Trap settings are configured to handle a defined DNS trap address not used in the environment and identify Internal DNS server IAW Threat Prevention best practices, which should correctly handle identification of actions to access a non-reputable address through a DNS query.

