Using PAC files with McAfee Client Proxy…or Not

The question is often asked: which takes precedence McAfee Client Proxy (MCP) or Proxy Auto Configuration (PAC) file? The answer is "yes!" 😉 **Technically the answer is MCP, but only if it is properly configured to do so.** A PAC file operates at the application layer, so a **PAC file has the first chance to alter an application's traffic flow, IF the application honors the PAC file.** MCP operates at the network level, so **MCP has the last chance to redirect traffic FOR ANY application** (whether proxy-aware or not), **IF it is configured to intercept the destination/port combination**. MCP can work with or without a PAC file and PAC files can work with or without MCP. With the multiple redirection options supported for MWG and WGCS (UCE and WPS) there is flexibility to use the appropriate methods for each system and any of their possible operating environments.

PAC files are **usually only required in environments that don't resolve external DNS and/or clients do not have a default Internet route.** They also **may be required if the logic for selecting the proper proxy is exceedingly complex** such that it cannot be supported by MCP Policy alone. PAC files are also useful for **handling systems that do not have MCP.** Lastly, combining use of MCP with PAC files can be helpful in **testing or transitioning to MVISION UCE in environments where PAC files are currently in use**.

The Basics of PAC files

PAC files (and WPAD, an automated distribution method for PAC files) are wonderful tools for working in explicit proxy environments and with applications that can be configured to use them. PAC files have been in use since 1996 and are well documented. PAC files operate at the application layer.

Pros

- Most flexible solution allows you to granularly determine what traffic should be directed to which proxies and what traffic should go direct
- Supports secure network environments where there is no external DNS resolution and no default route
- Most browsers are configured by default to automatically use a PAC file if the environment is configured for Web Proxy Auto Deployment (WPAD)
- Allows for granular proxy selection with different fallback options for each scenario and can even support intelligent load balancing to enhance caching performance of local proxy caches
- Supported by most browsers regardless of operating system
- Can be configured to failover, fail open, or fail closed
- Supports redirection of any port

Cons

- Only works for applications and TCP protocols that are PAC file aware and honor the PAC file
- If the PAC file cannot be reached on application start (e.g. captive portal environments) the browser or application will need to be restarted after the PAC file can be reached
- PAC file changes only get reflected when the application is restarted
- Easily bypassed or subverted unless there are compensating controls that may also impact operation in uncontrolled environments

- Does not pass any context about the client to the destination proxy
- PAC files can be complicated and difficult to maintain, syntax errors can break operation and it is easy to implement incorrect logic that results in unexpected operation
- PAC files used alone cannot transparently authenticate to a cloud proxy
- PAC files can not add encryption, some ISPs will block unencrypted proxy CONNECT requests
- Proxy selection cannot be configured for fastest response time
- Use of HTTP3/QUIC will bypass the PAC file unless the network blocks UDP on 443 and 80

The Basics of MCP

The MCP agent is also a wonderful redirection method for explicit proxy environments. Introduced by McAfee in 2014, MCP remains the most robust endpoint redirection agent available. The agent operates as a transparent web proxy for all applications. All vendor supported Windows and Mac operating system versions can utilize MCP. As previously stated, MCP operates at the network layer.

Pros

- Completely application agnostic
- Supplies prompt-less user and group information to the proxy without need for a directory connection or synch
- Highly tamper resistant, not easily bypassed, administrative controlled bypass and uninstall
- Allows for alternate proxy and bypass based on destination port, domain, IP, and process name
- Adds additional context for filtering decisions, policy name, process name, OS, OS version, system name, and more
- Can be configured to failover, fail open, or fail closed (when internet is available, but no proxies can be reached)
- Network aware, operation can be adjusted based on network location
- Redirection policy automatically updated on all clients within a few minutes of change
- When using cloud service will automatically select best proxy based on geolocation of client
- Can be used with Web Gateway Cloud Service and McAfee Web Gateway simultaneously
- Can intercept any configured port
- Proxy selection can be based on fastest response time or first available
- Can add encryption for unencrypted protocols
- MCP Policy can block HTTP3/QUIC so that this traffic doesn't bypass the proxy

Cons

- Requires installation of an agent that only runs on Windows and Mac operating systems
- Needs to have routing to a supported proxy (cloud, or on premise)
- Requires standard DNS resolution for domain-based redirection decisions
- MCP through version 4.2 only supports redirection of HTTP and HTTPS protocols
- Through 4.2 only supports redirection of IPv4 traffic (can block use of IPv6 to force use of IPv4)
- Through 4.2 only intercepts configured ports
- Selection of the optimal cloud proxy requires DNS resolution of McAfee cloud proxy domains

Con for Both PAC Files and MCP

If a client is enabled to use MCP and/or PAC file and is in an explicitly proxied third party network, MCP will stand down, the client will need use the PAC file of the local network, and the traffic will no longer be visible or controlled by the intended corporate controlled proxy.

Authentication, Redirection, Failover and Redundancy

There are many resources available that document the details of MCP and PAC file operation. Only a few highlights relevant to this discussion are repeated here.

Both PAC files and MCP can be used for redirection to McAfee Web Gateway, and McAfee Web Gateway Cloud Service. PAC files do not have any built-in authentication capabilities, but there are multiple standard authentication methods available for explicit proxy deployments. When using a PAC file to connect explicitly to a cloud proxy from an unknown or shared address, completely prompt-less and accurate authentication is not possible. If MCP is used, prompt-less authentication to any McAfee proxy is possible from any source address, shared or not.

PAC files, based on the implemented logic, ultimately return a string of the form “PROXY <proxy_1>:<port_1>; PROXY <proxy_2>:<port_2>; DIRECT” to the application. For example: “PROXY c123.saasprotection.com:8080; PROXY 10.0.1.10:8080; DIRECT” When the application sees this string, it will try to connect to the first listed proxy, if that connection fails, it will try to use the next listed proxy, and if that fails and DIRECT is included at the end of the string it will send the traffic direct on its original destination port. There is no limit to the number of proxies that can be put in the list, but usually just one or two are used and it is rare that more than 4 proxy entries are used. Proxies can be specified by name or IP and load balanced addresses can obviously be used.

MCP presently works off two proxy lists specified in the MCP policy. Proxies are specified by name or IP with the associated port. If a cloud proxy is specified, MCP will work with McAfee Global Routing manager to select the best (lowest latency) proxy for a given source location. The proxy lists can be configured independently and can work based on first available, or fastest response time. If the MCP policy designates traffic to the alternate proxy list and none of the configured alternate proxies are reachable, the traffic will be sent to the active primary proxy. If traffic is designated by policy for the primary proxy (not bypassed or designated specifically for alternate proxy) and none of the configured primary proxies are available, the traffic can be blocked or allowed direct based on policy.

Getting the Best of Both Worlds

As can be seen from the above, many of the advantages of one solution are disadvantages of the other. Systems using PAC files can coexist with systems using MCP and utilize the same services on the same network. In some situations, it may be advantageous to utilize both methods actively on the same system, or to utilize different methods in different network environments, and different systems. **MCP can work with a PAC file without alterations to the existing PAC file and without alterations to any browser settings!**

Things to keep in mind when using PAC files with MCP:

- MCP can easily forward proxied requests from an application using a PAC file **IF it is configured to intercept the proxy port AND the proxy address is not in the bypass list**
- When MCP forwards a proxied request, **bypasses and alternate proxy by domain and original destination IP will not work. Bypass and alternate proxy by process name and port will work.**
- **For sites that must be accessed directly, the PAC file is configured to send it DIRECT AND (MCP is configured to bypass by port (80,443), OR by destination IP, OR by process).**
- **For mobile systems using MCP, it is not advisable to bypass by ports 80 and 443 or by process unless you can have different MCP policies that can be applied when off net.**

MCP and PAC File Interactions

The PAC file can designate a return string that is customized based on application and client context. Commonly used proxy decision criteria are, destination host, destination domain, destination IP, and system source IP. Based on availability of any proxies listed in the string, the application will either send the traffic down the stack destined for a proxy, or destined DIRECT. If DIRECT is not listed in the return string and no proxies in the returned string can be reached, the traffic will not even be sent down the stack and the application will error.

MCP operates at the network level and must make its redirect decision at the time of the TCP SYN. For applications that follow a PAC file, MCP will see TCP SYN traffic designated DIRECT addressed to the destination server IPs on ports 80 and 443 and MCP will see traffic directed to proxy addresses on designated proxy ports. For applications that do not obey PAC files, TCP SYN traffic for HTTP and HTTPS will generally be seen by MCP as sent to the destination server IP on ports 80 and 443.

How MCP handles this traffic is based on configured policy which includes:

- Network awareness: MCP can be configured to completely stand down based on whether or not it can reach (make a TCP connection to any “landmark” designated by host:port)
- Ports configured for Intercept: 80 and 443 are intercepted by default. Proxy traffic using non-intercepted ports, will bypass MCP, even if MCP is configured to always redirect.
- Proxy lists: Alternate and Primary proxy lists can be independently configured as first available, or fastest response time. If none of the proxies in the alternate proxy list can be reached, the primary proxy will be used for all intercepted traffic. If the primary proxies cannot be reached, traffic destined for the primary proxy will bypass MCP. If an alternate proxy can be reached, any traffic designated for the alternate proxies will be redirected independently.
- Bypass lists: Bypass lists apply regardless of whether any systems in the proxy lists are available or not. Bypass can be by process name, destination port, domain name, or destination IP.
- Alternate proxy redirection lists: Alternate proxy redirection lists can be used based on process name, destination port, domain name, or destination IP.
- Non-redirected port blocking: For **ports that are not configured for redirection** there is the option to allow the traffic, block the traffic for all process names not in the process name bypass list, or block traffic for specific process names.

There are many configuration options and combinations for PAC and MCP to handle almost any use case. The options are extended when host firewall, VPN ACLs and network firewall configurations can

also influence operation and availability of the proxies configured in MCP and PAC file policy. McAfee Web Gateway also adds the possibility of hosting multiple PAC or WPAD files, and dynamically altering them before delivery based on requesting client characteristics including client source IP.

Use Cases Involving PAC files

PAC files are usually only required in environments that don't resolve external DNS. They also may be required if the logic for selecting the proper proxy is exceedingly complex such that it cannot be supported by MCP alone. **PAC files are also useful for handling systems that do not have MCP.** Lastly, combining use of MCP with PAC files can be helpful in testing or transitioning to MVISION UCE in environments where PAC files are currently in use.

Testing or Transitioning in Environments Where PAC Files are Currently in Use

MCP Config (Always Redirecting)

Intercept ports: 80, 443, and whatever proxy ports are currently used in PAC file

Primary proxy servers: FQDN cloud proxies first**,** local proxies next, first available

Alternate proxy servers: Local proxies

Alternate proxy redirection list: PAC File proxy ports and / or proxy addresses that should go to local proxies

Bypass lists: Local addresses (**except resolved proxy IP addresses**), and locations that go direct regardless of network

Block lists: Optional process name based for non-intercepted ports

VPN ACLs, Firewall, Routing, DNS and MWG

Split tunnel VPN. Allow reaching McAfee cloud addresses direct, allow direct sites that need to go direct when on the VPN

Firewall: Allow reaching McAfee cloud addresses direct on proxy ports (80, 8080, 8084)

DNS: must be able to at least resolve saasprotection.com (Cloud ePO managed WGCS) or mcafee-cloud.com (UCE managed WGCS) for availability SLA to apply. Hardcoded IPs could be used for testing or even production but the availability SLA would not apply.

Routing: must be able to route all McAfee WGCS addresses (trust.mcafee.com/web) and selected port must be allowed.

MWG: (Optional if local MWG to be used for some sites) Proxies on port 8080 MCP auth on 8080

A Simple and Common Use Case

- **On Net** use cloud proxies by default, some sites use local proxies, some sites go direct
- **On VPN** use cloud proxies by default, use local proxies for some sites, some sites go direct
- **Off Net** use cloud for all destinations except those designated direct

MCP only

MCP Config (Always Redirecting)

Intercept ports: 80, 443

Primary proxy servers: FQDN cloud proxies first**,** local proxies next, first available

Alternate proxy servers: Local proxies, then cloud proxy next, first available

Alternate proxy redirection list: Sites that need to use local proxies

Bypass lists: Local addresses, locations that go direct regardless of network

Block lists: Optional process name based for non-intercepted ports

VPN ACLs, Firewall, Routing, and DNS

Split tunnel VPN. Allow reaching McAfee cloud addresses direct, allow direct sites that need to go direct when on the VPN

Firewall: Allow reaching McAfee cloud addresses direct

DNS: must be able to resolve public and private addresses

Routing: must be able to route all McAfee WGCS addresses and selected port must be allowed.

MWG: Proxies on port 8080 MCP auth on 8080

Simple Use Case MCP Configuration in Action

- Direct.example.com – Should always go direct. (Bypass in MCP Bypass list)
- Localalways.example.com – should always go to local proxy when on net or on VPN unless local proxy cannot be reached in which case cloud proxy will be tried
- Any other site should go to cloud proxies unless they cannot be reached in which case local proxies will be tried.

PAC file only – Could be implemented but why?

If a complex PAC file is going to be implemented and maintained, it should be combined with MCP to get the benefits stated at the beginning of this article for controlled Windows and Mac systems.

Combined PAC file and MCP – PAC file covers machines without MCP

PAC file Config (Public Address is any address not in MCP IP bypass list)

Return DIRECT: Local addresses and any locations that need to go direct regardless of network.

PAC File: Return “FQDN Cloud Proxy:8080; LocalProxy:8080; DIRECT” for any sites/domains that should go to cloud proxy, if possible. All other sites return “Public Address:9090; Local Proxy: 8080; FQDN Cloud Proxy:8080; DIRECT”

MCP Config (Always Redirecting)

Intercept ports: 80, 443, 8080, 9090

Primary proxy servers: FQDN cloud proxies first**,** local proxies next, first available

Alternate proxy servers: Local proxies

Alternate proxy redirection list: Port 9090

Bypass lists: Local addresses, locations that go direct regardless of network

Block lists: Optional process name based for non-intercepted ports

VPN ACLs, Firewall, Routing, and DNS

Split tunnel VPN. Allow reaching McAfee cloud addresses direct, allow direct sites that need to go direct when on the VPN

Firewall: Allow reaching McAfee cloud addresses direct

DNS: must be able to at least resolve saasprotection.com (Cloud ePO managed WGCS) or mcafee-cloud.com (UCE managed WGCS) for SLA for availability to apply

Routing: must be able to route all McAfee WGCS addresses and selected port must be allowed.

MWG: Proxies on ports 8080, and 9090 (MCP auth on 8080 and 9090)

Simple Use Case MCP plus PAC Configuration in Action

- Direct.example.com – Should always go direct. (Bypass in MCP Bypass list and PAC file) PAC file returns “DIRECT”
- Localalways.example.com – should always go to local proxy when on net or on VPN unless local proxy cannot be reached in which case cloud proxy will be tried. PAC file returns “8.8.8.8:9090; mwg.domain.local:8080; c1234567890.saasprotection.com:8080; DIRECT”
- Any other site should go to cloud proxies unless they cannot be reached in which case local proxies will be tried.

Off Net:

PAC file not present. MCP will intercept and send to cloud proxy anything on port 80 and 443 that is not in bypass list. Local proxies cannot be reached. Direct.example.com in bypass list so it goes direct. Clients without MCP go direct for everything.

On Net or On VPN:

When accessing http://direct.example.com or https://direct.example.com, PAC file should always return DIRECT which will then be looked up through DNS. If MCP is present MCP will bypass and allow to go direct because direct.example.com was in bypass list.

When accessing localalways.example.com PAC file always returns “Public Address:9090; mwg.domain.local:8080; c1234567890.saasprotection.com:8080; DIRECT”. Port 9090 intercepted by MCP, Public Address not in bypass list, Port 9090 is in alternate proxy redirection list, directed to (alternate) local proxy by default. Clients without MCP will still “fail” to local proxy because they cannot reach Public Address:9090 as it won’t be redirected to a valid proxy. If the local proxy also cannot be reached the clients could “fail” to the cloud with IP based authentication

When accessing www.google.com PAC file returns “FQDN Cloud Proxy:8080; mwg.domain.local:8080; DIRECT” Port 8080 intercepted by MCP, Public Address not in bypass list, Port 8080 is not in alternate proxy redirection list, directed to (primary) cloud proxy by default. On premise clients without MCP will still use the cloud with IP based authentication. VPN clients without MCP traffic will try to go to cloud proxy, but if cloud proxy can be reached via split tunnel IP authentication won’t work, so it is recommended to either use MCP with all VPN clients, or **if VPN is present without MCP do not split tunnel the cloud proxy addresses.** This will allow VPN clients without MCP to authenticate via IP like on premise clients.

If VPN policy must be the same for clients with and without MCP, and the desire is to have a singular PAC file (at the expense of using the local proxies for all sites for clients without MCP) the default return string for the PAC file could be: “Public Address:8080; mwg.domain.local:8080; DIRECT”

Example PAC file (only one needed) for above Use Case:

```
function FindProxyForURL(url, host) {
  var stCloudProxy “c1234567890.saasprotection.com:8080; mwg.domain.local:8080; DIRECT”
  var stLocalProxy “8.8.8.8:9090; mwg.domain.local:8080; c1234567890.saasprotection.com:8080;
DIRECT”

  if (localHostOrDomainIs(host, "direct.example.com")) {
    return "DIRECT";
  }

  if (localHostOrDomainIs(host, "localalways.example.com")) {
    return stLocalProxy;
  }

  return "stCloudProxy";
}
```

A Simple and Less Common Use Case

- **On net** use local proxies by default, use cloud proxies for certain sites, some sites go direct
- **On VPN** use local proxies by default, use cloud proxies for certain sites, some sites go direct
- **Off net** use cloud for all destinations except those designated direct

MCP only

MCP Config (Always Redirecting)

Intercept ports: 80, 443

Primary proxy servers: Local proxies, then cloud proxy next, first available

Alternate proxy servers: FQDN cloud proxies first**,** local proxies next, first available

Alternate proxy redirection list: Sites that need to use cloud proxies

Bypass lists: Local addresses, locations that go direct regardless of network

Block lists: Optional process name based for non-intercepted ports

VPN ACLs, Firewall, Routing, and DNS

Split tunnel VPN. Allow reaching McAfee cloud addresses direct, allow direct sites that need to go direct when on the VPN

Firewall: Allow reaching McAfee cloud addresses direct

DNS: must be able to resolve public and private addresses

Routing: must be able to route all McAfee WGCS addresses and selected port must be allowed.

MWG: Proxies on port 8080 MCP auth on 8080

Simple Use Case - MCP Configuration in Action

- Direct.example.com – Should always go direct. (Bypass in MCP Bypass list)
- Cloudalways.example.com – should always go to cloud proxy (PAC file returns "c1234567890.saasprotection.com:8080; mwg.domain.local:8080; DIRECT")
- Other traffic goes to local proxies, and "fails" to cloud proxies.

PAC file only – Could be implemented but why?

If a complex PAC file is going to be implemented and maintained, it should be combined with MCP to get the benefits stated at the beginning of this article for controlled Windows and Mac systems.

Combined PAC file and MCP – PAC file covers machines without MCP

PAC file Config (Public Address is any address not in MCP IP bypass list)

Return DIRECT: Local addresses and any locations that need to go direct regardless of network.

PAC File: Return “Cloud Proxy:8080; mwg.domain.local:8080; DIRECT” for any sites/domains that should go to cloud proxy. Other sites return “Public Address:9090; Local Proxy: 8080; FQDN Cloud Proxy:8080; DIRECT”

Off Net:

PAC file not present. MCP intercepts and sends port 80 and 443 traffic that is not in the bypass list to cloud. Direct.example.com is bypassed and goes direct. Clients without MCP send everything direct.

On Net or On VPN:

When accessing http://direct.example.com or https://direct.example.com, PAC file should always return DIRECT which will then be looked up through DNS. If MCP is present, MCP will bypass and allow to go direct because direct.example.com was in bypass list.

When accessing cloudalways.example.com PAC file always returns “FQDN Cloud Proxy:8080; mwg.domain.local:8080; DIRECT”. Port 8080 intercepted by MCP, FQDN Cloud Proxy not in bypass list, Port 8080 is not in alternate proxy redirection list, directed to (primary) cloud proxy by default. On premise clients without MCP will use the cloud with IP based authentication. For VPN clients without MCP, the traffic will try the cloud proxy. If the cloud proxy is reachable via split tunnel IP authentication won’t work, so it is recommended to **use MCP with all VPN clients, or if VPN is present without MCP do not split tunnel the cloud proxy addresses for those clients.** This will allow VPN clients without MCP to authenticate via IP like on premise clients.

If VPN policy must be the same for clients with and without MCP, and the desire is to have a singular PAC file (at the expense of using the local proxies for all sites for clients without MCP) the “cloud” return string for the PAC file could be: “Public Address:8080; mwg.domain.local:8080; DIRECT”

When accessing www.google.com PAC file returns “Public Address:9090; mwg.domain.local:8080; DIRECT” Port 9090 intercepted by MCP, Public Address not in bypass list, Port 9090 is in alternate proxy redirection list, directed to (alternate) local proxy by default. Clients without MCP will use the local proxy because they cannot reach Public Address:9090 as it won’t be redirected to a valid proxy.

Example PAC file (only one needed) for above Use Case:

```
function FindProxyForURL(url, host) {
  var stCloudProxy “c1234567890.saasprotection.com:8080; mwg.domain.local:8080; DIRECT”
  var stLocalProxy “8.8.8.8:9090; mwg.domain.local:8080; DIRECT”

  if (localHostOrDomainIs(host, "direct.example.com")) {
    return "DIRECT";
  }

  if (localHostOrDomainIs(host, "cloudalways.example.com")) {
    return stCloudProxy
  }

  return "stLocalProxy";
}
```

Most Complex Use Case – Requires Using MCP and PAC Files

- **On Net** use corp proxies by default, use cloud for certain sites, except those designated direct
- **On VPN** use corp proxies for certain sites, use cloud by default, except those designated direct
- **Off Net** use cloud for all destinations except those designated direct

Directly below is recommended for maximum resiliency and least complexity.

Multiple PAC Files, Single MCP Policy

PAC file Configs (Public Address is any address not in MCP IP bypass list)

A common PAC file cannot accurately determine client network location using myipAddress. Matching your corporate network via myipAddress would not be reliable as a VPN client on a network with private addresses that overlap your corporate addresses would be detected as on network**. Better is to serve a different PAC file dependent on the source address of the client (on net or on VPN) hitting the PAC file server on the corporate network.** This is possible if MWG is serving the PAC file.

Return DIRECT: Local addresses and any locations that need to go direct regardless of network.

On net PAC File: Return "Public Address:8080; LocalProxy:8080; DIRECT" for any sites/domains that should go to cloud proxy, if possible. All other sites return "Public Address:9090; LocalProxy: 8080; DIRECT"

On VPN PAC file: Return "Public Address:9090; LocalProxy:8080 DIRECT" for any sites/domains that should go to corp proxies when on VPN. All other sites return "Public Address:8080; LocalProxy: 8080; DIRECT"

MCP Config (Always Redirecting)

Intercept ports: 80, 443, 8080, 9090

Primary proxy servers: FQDN cloud proxies first**,** local proxies next, first available

Alternate proxy servers: Local proxies

Alternate proxy redirection list: Port 9090

Bypass lists: Local addresses, locations that go direct regardless of network

Block lists: Optional process name based for non-intercepted ports

VPN ACLs, Firewall, Routing, and DNS

Split tunnel VPN. Allow reaching McAfee cloud addresses direct, allow direct sites that need to go direct when on the VPN

Firewall: Allow reaching McAfee cloud addresses direct

DNS: must be able to at least resolve saasprotection.com (Cloud ePO managed WGCS) or mcafee-cloud.com (UCE managed WGCS) for SLA for availability to apply

Routing: must be able to route all McAfee WGCS addresses and selected port must be allowed.

MWG: Proxies on ports 8080, and 9090 (MCP auth on 8080 and 9090)

Complex Use Case Configuration in Action

- DirectAlways.example.com Should go direct always. (Bypass in both MCP Bypass lists and PAC files)
- DirectVPN.example.com – Should go direct only when on VPN. (Bypass in On VPN MCP Bypass list and PAC files)
- DirectOnNet.example.com – should go direct only when on Net. (Bypass in On Net MCP Bypass list and PAC files)
- CloudAlways.example.com – should always go to cloud proxy (PAC file returns "FQDN Cloud Proxy:8080; LocalProxy:8080; DIRECT" regardless of client address)
- LocalAlways.example.com – should always go to local proxy when on net or on VPN (PAC file returns "Public Address:9090; LocalProxy:8080; DIRECT" regardless of client address)
- Vpn.example.com – should go to cloud proxy when off net, or on VPN, should go to local proxy when on net. (The on vpn PAC file returns "FQDN Cloud Proxy:8080; LocalProxy:8080; DIRECT"; the on net PAC file returns "Public Address:9090; LocalProxy:8080; DIRECT")
- Onnet.example.com – should go to cloud proxy when on net, or offnet, local proxy when on the vpn. (The on net PAC file returns "FQDN Cloud Proxy:8080; LocalProxy:8080; DIRECT"; the on net PAC file returns "Public Address:9090; LocalProxy:8080; DIRECT")
- Any other site should go to cloud when off net or on vpn and to local proxy if on net. (Default PAC file return for on vpn PAC file: "FQDN Cloud Proxy:8080; LocalProxy:8080; DIRECT"; Default PAC file return for the on net PAC file: "Public Address:9090; LocalProxy:8080; DIRECT")

Offnet:

PAC file not present. MCP will intercept and send to cloud proxy anything on port 80 and 443 that is not in bypass list. DirectAlways.example.com, DirectVPN.example.com and DirectOnNet.example.com are in the MCP bypass list so they go direct.

On VPN:

When accessing DirectAlways.example.com or DirectVPN.example.com, PAC file should return DIRECT which will then be looked up through DNS. MCP (if present) will bypass and allow to go direct because Direct.example.com and DirectVPN.example.com were in the bypass list. DirectOnNet.example.com is also in the MCP bypass list but the On VPN PAC file sends that to the proxy by default, so it gets intercepted.

When accessing cloudalways.example.com On VPN PAC file returns "PublicAddress:8080; LocalProxy:8080; DIRECT" Port 8080 intercepted by MCP, Public Address not in bypass list, and port 8080 is not in alt proxy redirection list, directed by MCP to primary cloud proxy (cloud) by default.

When accessing localalways.example.com PAC file always returns "Public Address:9090; LocalProxy:8080; DIRECT" Port 9090 intercepted by MCP, Public Address not in bypass list, Port 9090

is in alternate proxy redirection list, directed to (alternate) local proxy by default. If MCP is not present Public Address:9090 will not connect and PAC file will fail to LocalProxy:8080.

When accessing vpn.example.com PAC file returns “Public Address:9090; LocalProxy:8080; DIRECT” Port 9090 intercepted by MCP, Public Address not in bypass list, Port 9090 is in alternate proxy redirection list, directed to (alternate) local proxy by default. If MCP is not present, Public address:9090 will not connect and PAC file will fail to LocalProxy:8080.

When accessing onnet.example.com PAC file returns “Public Address:8080; LocalProxy:8080; DIRECT” Port 8080 intercepted by MCP, Public Address not in bypass list, Port 8080 is not in alternate proxy redirection list, directed to (primary) cloud proxy by default.

When accessing www.google.com PAC file returns “Public Address:8080; LocalProxy:8080; DIRECT” Port 8080 intercepted by MCP, Public Address not in bypass list, Port 8080 is not in alternate proxy redirection list, directed to (primary) cloud proxy by default.

Without MCP, any traffic designated for Public Address:8080 or 9090 will “fail” to the local proxy, so **it is recommended that any VPN clients also be equipped with MCP so the traffic is not backhauled through the local proxy**.

Example On VPN PAC file for above Use Case:

```
function FindProxyForURL(url, host) {
  var stCloudProxy “8.8.8.8:8080; mwg.domain.local:8080; DIRECT”
  var stLocalProxy “8.8.8.8:9090; mwg.domain.local:8080; DIRECT”

  if (localHostOrDomainIs(host, "DirectAlways.example.com") ||
    localHostOrDomainIs(host, "DirectOnVPN.example.com")) {
    return "DIRECT";
  }

  if (localHostOrDomainIs(host, "vpn.example.com") ||
    localHostOrDomainIs(host, "localalways.example.com")) {
    return stLocalProxy;
  }

  return "stCloudProxy";
}
```

On Network:

When accessing DirectAlways.example.com or DirectOnNet.example.com, On Net PAC file should return DIRECT which will then be looked up through DNS. MCP (if present) will bypass and allow to go direct because Direct.example.com and DirectOnNet.example.com were in the bypass list. DirectVPN.example.com is also in the MCP bypass list but the On Net PAC file sends that to the proxy by default, so it gets intercepted and proxied.

When accessing cloudalways.example.com PAC file always returns “Cloud Proxy:8080; LocalProxy:8080; DIRECT” Port 8080 intercepted by MCP, Cloud Proxy not in bypass list, and port

8080 is not in alt proxy redirection list, directed to primary cloud proxy (cloud) by default. If MCP is not present, traffic will be proxied in the cloud and will be authenticated via source IP.

When accessing localalways.example.com PAC file always returns “Public Address:9090; LocalProxy:8080; DIRECT” Port 9090 intercepted by MCP, Public Address not in bypass list, Port 9090 is in alternate proxy redirection list, directed to (alternate) local proxy by default. If MCP is not present, Public Address:9090 will not connect and PAC file will fail to LocalProxy:8080.

When accessing vpn.example.com PAC file returns “Public Address:9090; LocalProxy:8080; DIRECT” Port 9090 intercepted by MCP, Public Address not in bypass list, Port 9090 is in alternate proxy redirection list, directed to (alternate) local proxy by default.

When accessing onnet.example.com PAC file returns “Cloud Proxy:8080; LocalProxy:8080; DIRECT” Port 8080 intercepted by MCP, Cloud Proxy not in bypass list, Port 8080 is not in in alternate proxy redirection list, directed to (primary) cloud proxy by default.

When accessing www.google.com PAC file returns “Public Address:9090; LocalProxy:8080; DIRECT” Port 9090 intercepted by MCP, Public Address not in bypass list, Port 9090 is in alternate proxy redirection list, directed to (alternate) local proxy by default.

If MCP is not present, traffic destined for the cloud proxy will be proxied in the cloud and will be authenticated via source IP.

Example On Network PAC file for above Use Case:

```
function FindProxyForURL(url, host) {
  var stCloudProxy “c1234567890.saasprotection.com:8080; mwg.domain.local:8080; DIRECT”
  var stLocalProxy “8.8.8.8:9090; mwg.domain.local:8080; DIRECT”

  if (localHostOrDomainIs(host, "DirectAlways.example.com") ||
    localHostOrDomainIs(host, "DirectOnNet.example.com")) {
    return "DIRECT";
  }

  if (localHostOrDomainIs(host, "onnet.example.com") ||
    localHostOrDomainIs(host, "cloudalways.example.com”)) {
    return stCloudProxy
  }

  return "stLocalProxy";
}
```

Another Way to Implement the Complex Use Case – Single PAC file, Multiple MCP Policies

If it is not possible to serve different PAC files based on location, it may be possible to utilize a single PAC file and deliver a different MCP policy based on network location. This community article describes how applying different MCP policies might be achieved on Windows using a script, a couple files and Advanced Scheduler: https://community.mcafee.com/t5/Web-Gateway-Cloud-Service/How-Can-MCP-Policy-be-Changed-Dynamically-Based-on-Network/m-p/682854#M140 **This**

solution was not recommended because it makes it much harder to maintain the MCP Policies. And MCP policy switching may be difficult to implement. It is described here to show additional configuration flexibility.

- **On Net** use corp proxies by default, use cloud for certain sites, except those designated direct
- **On VPN** use corp proxies for certain sites, use cloud by default, except those designated direct
- **Off Net** use cloud for all destinations except those designated direct

PAC file Config (Public Address is any address not in MCP IP bypass list)

Return DIRECT: Local addresses and any locations that need to go direct regardless of network.

PAC File: For the default return string **r**eturn "Public Address 1:8080; LocalProxy:8080; DIRECT" these will go to local proxy when on net and cloud proxy when offnet. Return "Public Address 1:9090; LocalProxy: 8080; DIRECT" for any sites/domains that should go to cloud proxy when on net and local proxy when off net. Return "Public Address 2:9091; LocalProxy: 8080; DIRECT" for any sites/domains that should always go to the local proxies. Return "Public Address 3:9091; LocalProxy: 8080; DIRECT" for any sites/domains that should always go to the cloud proxies.

On Net MCP Config (Always Redirecting)

Intercept ports: 80, 443, 8080, 9090,9091

Primary proxy servers: Local proxies

Alternate proxy servers: FQDN cloud proxies first**,** local proxies next, first available

Alternate proxy redirection list: Port 9090, IP Address 4.4.4.4

Bypass lists: Local addresses, locations that go direct when On Net

Block lists: Optional process name based for non-intercepted ports

On VPN MCP Config (Always Redirecting also covers Off Net)

Intercept ports: 80, 443, 8080, 9090,9091

Primary proxy servers: FQDN cloud proxies first**,** local proxies next, first available

Alternate proxy servers: Local proxies

Alternate proxy redirection list: Port 9090, IP Address 1.1.1.1

Bypass lists: Local addresses, locations that go direct when On VPN

Block lists: Optional process name based for non-intercepted ports

VPN ACLs, Firewall, Routing, MWG, and DNS

Split tunnel VPN. Allow reaching McAfee cloud addresses direct, allow direct sites that need to go direct when on the VPN

Firewall: Allow reaching McAfee cloud addresses direct

DNS: must be able to at least resolve saasprotection.com (Cloud ePO managed WGCS) or mcafee-cloud.com (UCE managed WGCS) for SLA for availability to apply

Routing: must be able to route all McAfee WGCS addresses and selected port must be allowed.

MWG: Proxies on ports 8080, and 9090 (MCP auth on 8080 and 9090)

Complex Use Case Alternate Configuration in Action

- DirectAlways.example.com Should go direct always. (Bypass in both MCP Bypass lists and PAC files) PAC file returns DIRECT, DNS lookup happens, bypassed by both MCP policies, always goes DIRECT.
- DirectVPN.example.com – Should go direct only when on VPN. (Bypass in On VPN MCP Bypass list and PAC files) PAC file returns DIRECT, DNS lookup happens, bypassed by On VPN MCP policy, goes DIRECT only if On VPN. If On network with MCP, it is intercepted and sent to the local proxy unless it is in the On Net MCP policy alternate proxy list.
- DirectOnNet.example.com – Should go direct only when on Net. (Bypass in On Net MCP Bypass list and PAC files) PAC file returns DIRECT, DNS lookup happens, bypassed by On Net MCP policy, goes DIRECT only if On Net. If On VPN with MCP, it is intercepted and sent to the cloud proxy unless it is in the On VPN MCP policy alternate proxy list.
- CloudAlways.example.com – should always go to cloud proxy (PAC file returns "Public Proxy 3:9091; mwg.domain.local:8080; DIRECT") On Net MCP Policy sends to Alternate proxy (cloud) because Public Proxy 3 is in On Net MCP Alternate proxy list. On VPN Policy sends to Primary Proxy (cloud) because Public Proxy 3 is not in the Alternate proxy list. Public Proxy 3 and port 9091 are not in either policy's bypass list.
- LocalAlways.example.com – should always go to local proxy (PAC file returns "Public Proxy 2:9091; mwg.domain.local:8080; DIRECT") On VPN MCP Policy sends to Alternate proxy (local) because Public Proxy 2 is in On VPN MCP Alternate proxy list. On Net Policy sends to Primary Proxy (local) because Public Proxy 2 is not in the Alternate proxy list. Public Proxy 2 and port 9091 are not in either policy's bypass list.
- PrimaryAlways.example.com – should always go to cloud proxy when on vpn or off net and go to local proxy when on net (PAC file returns "Public Proxy 1:8080; LocalProxy:8080; DIRECT") Public Proxy 1 and Port 8080 are not in the alternate proxy redirection lists and are not in the bypass list for either policy, so the traffic will go to the primary proxy (cloud for VPN Policy, local for On Net policy)
- AlternateAlways.example.com – should always go to cloud when on or off net, local proxy when on VPN (PAC file returns "Public Proxy 1:9090; mwg.domain.local:8080; DIRECT") Both policies have port 9090 in alternate proxy redirection list and Public Proxy 1 is not in the bypass list for either policy so traffic goes to the alternate proxy (local for VPN Policy, cloud for On Net policy)
- Vpn.example.com – should go to cloud proxy when off net, or on VPN, should go to local proxy when on net. (The PAC file returns "Public Proxy 1:8080; LocalProxy:8080; DIRECT") Public Proxy 1 and Port 8080 are not in the alternate proxy redirection lists and are not in the bypass list for either policy, so the traffic will go to the primary proxy (cloud for VPN Policy, local for On Net policy)

- OnNet.example.com – should go to cloud proxy when on net, or offnet, local proxy when on the vpn. (PAC file returns “Public Proxy 1:9090; mwg.domain.local:8080; DIRECT”) Port 9090 is in alternate proxy redirection list for both policies and is not in the bypass list for either policy, so traffic goes to the alternate proxy (local for VPN Policy, cloud for On Net policy)
- Any other site should go to cloud when off net or on vpn and to local proxy if on net. (Default PAC file return “Public Proxy 1:8080; LocalProxy:8080; DIRECT”) Public Proxy 1 and Port 8080 are not in the alternate proxy redirection list and are not in the bypass list for either policy so the traffic will go to the primary proxy (cloud for VPN, local for On Net)

Clients that use the PAC file and don’t have MCP will send all traffic to the local proxy with the PAC file configuration shown below. **It is recommended that all VPN clients also utilize MCP if this is not the desired behavior or look to the next example PAC file, for the use case above with two MCP Policies.**

Example PAC file for above Use Case:

```
function FindProxyForURL(url, host) {
  var stPrimaryProxy “8.8.8.8:8080; mwg.domain.local:8080; DIRECT”
  var stAlternateProxy “8.8.8.8:9090; mwg.domain.local:8080; DIRECT”
  var stLocalProxy “1.1.1.1:9091; mwg.domain.local:8080; DIRECT”
  var stCloudProxy “4.4.4.4:9091; mwg.domain.local:8080; DIRECT”

  if (localHostOrDomainIs(host, "DirectAlways.example.com") ||
    localHostOrDomainIs(host, "DirectVPN.example.com") ||
    localHostOrDomainIs(host, "DirectOnNet.example.com") {
    return "DIRECT";
  }

  if (localHostOrDomainIs(host, "AlternateAlways.example.com")) ||
    localHostOrDomainIs(host, "OnNet.example.com") {
    return stAlternateProxy
  }

  if (localHostOrDomainIs(host, "LocalAlways.example.com")) {
    return stLocalProxy
  }

  if (localHostOrDomainIs(host, "CloudAlways.example.com")) {
    return stCloudProxy
  }

  return "stPrimaryProxy";
}
```

With the PAC file below, clients that use the PAC file and don't have MCP will try to send some traffic to the cloud proxies (stAlternateProxy, and stCloudProxy return strings) and other traffic to the local proxies (stPrimaryProxy and stLocalProxy return strings). **All VPN clients should utilize MCP, or if MCP is not present, they should not be permitted to split tunnel to the cloud proxies. Otherwise VPN clients will fail when they get either the stCloudProxy or stLocalProxy return strings.** (This is because without MCP but with split tunnel, they will be able to make a TCP connection to the cloud proxy but the requests will be refused, because MCP authentication will not be present, and IP authentication won't work because the client will be coming from an arbitrary IP address.

Example Alternate PAC file for above Use Case:

```
function FindProxyForURL(url, host) {
  var stPrimaryProxy "8.8.8.8:8080; mwg.domain.local:8080; DIRECT"
  var stAlternateProxy "8.8.8.8:9090; c1234567890.saasprotection.com:8080;
      mwg.domain.local:8080; DIRECT"
  var stLocalProxy "1.1.1.1:9091; mwg.domain.local:8080; DIRECT"
  var stCloudProxy "4.4.4.4:9091; c1234567890.saasprotection.com:8080; mwg.domain.local:8080;
      DIRECT"

  if (localHostOrDomainIs(host, "DirectAlways.example.com") ||
    localHostOrDomainIs(host, "DirectVPN.example.com") ||
    localHostOrDomainIs(host, "DirectOnNet.example.com") {
    return "DIRECT";
  }

  if (localHostOrDomainIs(host, "AlternateAlways.example.com")) ||
    localHostOrDomainIs(host, "OnNet.example.com") {
    return stAlternateProxy
  }

  if (localHostOrDomainIs(host, "LocalAlways.example.com")) {
    return stLocalProxy
  }

  if (localHostOrDomainIs(host, "CloudAlways.example.com")) {
    return stCloudProxy
  }

  return "stPrimaryProxy";
}
```

There Are Other Ways to Implement the Complex Use Case

The above solutions were devised without need to use firewall rules or router ACLs to restrict proxy access in various network environments. These additional options add flexibility but also complexity and increase the probability of something getting unintentionally broken because different security groups or network groups were responsible for other aspects of the solution.