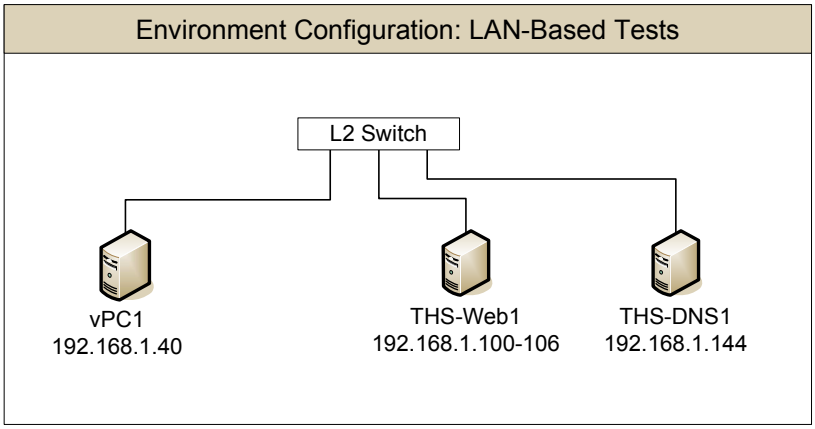
Experiment Configuration and Results

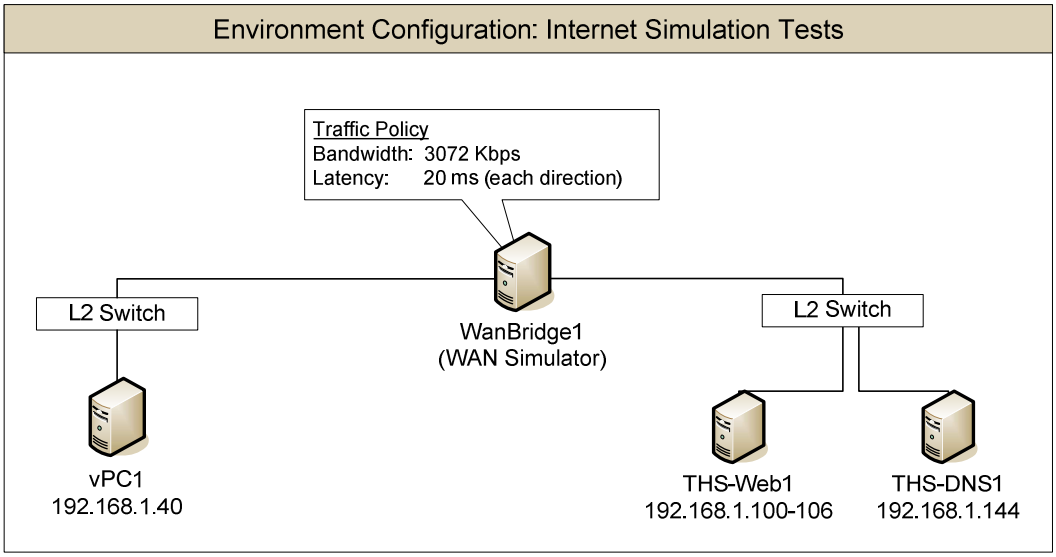
Environment Configuration

Servers

Hostname	IP Address	OS	Function
vPC1	192.168.1.40	Linux (OpenSuSE 11.4 x86-64)	workstation
THS-DNS1	192.168.1.144	Linux (OpenSuSE 11.4 x86-x64)	Authoritative Nameserver (Bind)
THS-Web1	192.168.1.100	Linux (OpenSuSE 11.4 x86-x64)	HTTP/HTTPS Server (Apache2)
WanBridge1	N/A	Linux (2.6.x Generic, x86-x32)	WAN simulator
THS-DNSResolver	192.168.1.145	Linux (OpenSuSE 11.4 x86-x64)	Recursive Nameserver (Bind)

Network Diagrams





Webserver Virtual Hosts

IP Address	VirtualHost Host Header	Certificate CN and SHA1 Hash	Cert Match
192.168.1.101	cpftest.covertpacket.com	cpftest.covertpacket.com a17e5dc70a7c31e1e6b819d450f0646fe33d7f81	***Yes***
192.168.1.102	cpftest.covertpacket.com	*.covertpacket.com 184eee13a42e63efdd08d663a03a7fb6e6244192	***Yes***
192.168.1.103	cpftest.covertpacket.com	mail.covertpacket.com cce9cdb442743559bc8116981af6c339e2c772bb	***No***
192.168.1.104	mail.covertpacket.com	mail.covertpacket.com cce9cdb442743559bc8116981af6c339e2c772bb	***Yes***
192.168.1.105	badguy.covertpacket.com	cpftest.covertpacket.com a17e5dc70a7c31e1e6b819d450f0646fe33d7f81	***No***
192.168.1.106	www.randomdomain.com	*.covertpacket.com 184eee13a42e63efdd08d663a03a7fb6e6244192	***No***

Certificate Hashes

Certificate SHA1 Hash	Common Name (CN)
a17e5dc70a7c31e1e6b819d450f0646fe33d7f81	cpftest.covertpacket.com (CN)
184eee13a42e63efdd08d663a03a7fb6e6244192	*.covertpacket.com (CN)
cce9cdb442743559bc8116981af6c339e2c772bb	mail.covertpacket.com (CN) smtp1.covertpacket.com (SAN) smtp2.covertpacket.com (SAN) smtp3.covertpacket.com (SAN) smtp4.covertpacket.com (SAN)

Experiment Configuration and Results

Test 1

Purpose
Demonstrate that the "-" qualifier is properly interpreted and enforced per RFC

Experiment Configuration
URL: https://cpftest.covertpacket.com
Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 -all"

Assertion:
CPF Result: Fail
CPF Action: Block
Reason: Directive "-all" will result in Fail action

Analysis
The CPF record associated with cpftest.covertpacket.com has only one directive, "-all". This means that the client application should interpret this as a CPF "Fail" result and block any given certificate – whether or not it is signed by a trusted CA. The client application blocked the connection as expected, and with negligible delay (avg 6 ms – 104 ms) and network overhead (443 bytes).

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.41	5.60	5.64	5.55
Time Δ (ms) - ACL Parse	0.67	0.52	0.62	0.60
Time Δ (ms) - Total	81.23	74.32	66.92	74.16
DNS bandwidth (bytes)	443	443	443	443
# of queries	2	2	2	2
CPF Result	Fail	Fail	Fail	
CPF Action	Block	Block	Block	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	99.79	98.81	96.66	98.42
Time Δ (ms) - ACL Parse	2.74	10.49	2.88	5.37
Time Δ (ms) - Total	184.21	179.46	174.72	179.46
DNS bandwidth (bytes)	443	443	443	443
# of DNS queries (count)	2	2	2	2
CPF Result	Fail	Fail	Fail	
CPF Action	Block	Block	Block	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
INIT - Host and Cert analysis

<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>
CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      -     all   NULL  primary     cpftest.covertpacket.com

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>
<===============  ClientApp ACL Inspection STARTS HERE  ===============>

ACL>  Processing ace #0:
ACL>  MATCH - Peer cert matches "all" mechanism with qualifier "-" --> Fail

<===============  ClientApp ACL Inspection ENDS HERE  ===============>

CPF result is [fail]
```

Test 2

Purpose

Demonstrate that the "~" qualifier is properly interpreted and enforced per RFC

Experiment Configuration

URL: https://cpftest.covertpacket.com
Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 ~all"

Assertion:

CPF Result: SoftFail
CPF Action: Warn
Reason: Directive "~all" will result in a SoftFail, and the browser will warn the user and prompt for approval in order to continue with downloading content.

Analysis

The client application matched on the "all" parameter and correctly interpreted the "~" to enforce the SoftFail result. The end user was warned about the questionable trust, as expected. The delta time (6 – 105 ms) and data exchange (443 bytes) were comparable to Test 1, which was expected since the record data is nearly the same.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.85	5.77	5.41	5.68
Time Δ (ms) - ACL Parse	0.61	0.54	0.30	0.48
Time Δ (ms) - Total	70.37	68.29	65.76	68.14
DNS bandwidth (bytes)	443	443	443	443
# of queries	2	2	2	2
CPF Result	SoftFail	SoftFail	SoftFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	99.18	95.44	98.44	97.69
Time Δ (ms) - ACL Parse	9.23	5.52	8.64	7.80
Time Δ (ms) - Total	177.96	174.62	173.78	175.45
DNS bandwidth (bytes)	443	443	443	443
# of queries	2	2	2	2
CPF Result	SoftFail	SoftFail	SoftFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-TIMER> Starting perl timer
CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 ~all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "~all"
CPF-PARSE> Qualifier: ~; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      ~     all   NULL  primary     cpftest.covertpacket.com

ACL>  Processing ace #0:
ACL>  MATCH - Peer cert matches "all" mechanism with qualifier "~" -->
SoftFail

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  Processing ace #0:
ACL>  MATCH - Peer cert matches "all" mechanism with qualifier "~" -->
SoftFail

<==============  ClientApp ACL Inspection ENDS HERE  ==============>
CPF result is [softfail]
```

Test 3

Purpose

Demonstrate that the default "~all" directive is enforced when a CPF record is returned but no directives are matched.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1"

Assertion:

CPF Result: SoftFail
CPF Action: Warn
Reason: Since no directives are provided, the default result of "SoftFail" should be enforced per RFC.

Analysis

The CPF interpreter identifies the record above as valid because it contains the version tag, but since there are no directives it has no explicit entries to match on. Thus, the default CPF result of "SoftFail" is enforced.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.72	5.59	5.57	5.63
Time Δ (ms) - ACL Parse	0.10	0.10	0.14	0.11
Time Δ (ms) - Total	66.96	66.84	78.32	70.71
DNS bandwidth (bytes)	438	438	438	438
# of queries	2	2	2	2
CPF Result	SoftFail	SoftFail	SoftFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	96.74	95.69	99.34	97.26
Time Δ (ms) - ACL Parse	0.36	0.36	0.46	0.39
Time Δ (ms) - Total	175.48	165.60	185.14	175.41
DNS bandwidth (bytes)	438	438	438	438
# of queries	2	2	2	2
CPF Result	SoftFail	SoftFail	SoftFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on  cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1"

CPF-PARSE> CPF version detected as "1" (originally "v=1")

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  MATCH DEFAULT - no explicit rules match, enforcing implicit rule
"SoftFail"

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

CPF result is [softfail]
```

Test 4

Purpose

Demonstrate that the "+" qualifier is properly interpreted and enforced per RFC

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
CPF records text (DNS): cpftest.covertpacket.com → "v=1 +all"

Assertion:

CPF Result: Pass
CPF Action: Allow
Reason: The "all" mechanism will match any certificate signature, and the "+" indicates that the connection should be permitted.

Analysis

The connection was permitted as anticipated, after matching on "+all".

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.75	5.64	5.48	5.62
Time Δ (ms) - ACL Parse	0.65	0.63	0.44	0.57
Time Δ (ms) - Total	66.86	66.90	72.13	68.63
DNS bandwidth (bytes)	443	443	443	443
# of queries	2	2	2	2
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	95.47	99.26	96.47	97.07
Time Δ (ms) - ACL Parse	2.78	2.98	2.57	2.78
Time Δ (ms) - Total	170.09	170.10	179.32	173.17
DNS bandwidth (bytes)	443	443	443	443
# of queries	2	2	2	2
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]
```

```
CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 +all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "+all"
CPF-PARSE> Qualifier: +; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     all   NULL  primary     cpftest.covertpacket.com


<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  MATCH - Peer cert matches "all" mechanism with qualifier "+" --> Pass

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

TIMER>      Received CPF Check answer at 67 ms
TIMER>      Full CPF process completed in 66.86 ms

CLOSE - CPF Checker
====================

CPF result is [pass]
```

Test 5

Purpose

Demonstrate that the hostname from the certificate common-name is used for the CPF lookup when a certificate hostname mismatch occurs.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: mail.covertpacket.com (CN)
smtp1.covertpacket.com (SAN)
smtp2.covertpacket.com (SAN)
smtp3.covertpacket.com (SAN)
smtp4.covertpacket.com (SAN)

Certificate Hash: (sha1) cce9cdb442743559bc8116981af6c339e2c772bb

Hostname mismatch = YES

CPF records text (DNS): mail.covertpacket.com → "v=1 +all"
cpftest.covertpacket.com → "v=1 -all"

Assertion:

CPF Result: Pass
CPF Action: Allow
Reason: CPF should detect the hostname mismatch and inspect the CPF record for the certificate common name, mail.covertpacket.com, instead of the hostname in the URL.

Analysis

The application initiated the CPF lookup for mail.covertpacket.com and applied the "+all" directive to permit the connection. If it hostname in the URL and certificate had both matched as "cpftest.covertpacket.com", then the CPF lookup for this hostname would have enforced the "-all" directive instead.

The CPF result of "pass" indicates that the service owner permits the certificate to be used for the hostname, but does not necessarily imply that the application should permit it. The hostname mismatch error will typically cause a real application to generate a warning.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.56	5.76	5.40	5.57
Time Δ (ms) - ACL Parse	0.56	0.57	0.52	0.55
Time Δ (ms) - Total	68.84	69.31	69.02	69.06
DNS bandwidth (bytes)	431	431	431	431
# of queries	2	2	2	2
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	96.53	98.20	95.15	96.63
Time Δ (ms) - ACL Parse	0.44	1.74	1.54	1.24
Time Δ (ms) - Total	169.35	172.57	168.66	170.19
DNS bandwidth (bytes)	431	431	431	431
# of queries	2	2	2	2
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
INIT - Host and Cert analysis

URL = https://cpftest.covertpacket.com
domain = covertpacket.com
hostname = cpftest.covertpacket.com
cpftest.covertpacket.com resolves to 192.168.1.103

Retrieving certificate attributes...
  > common name = mail.covertpacket.com
  > subject alt name = smtp4.covertpacket.com
  > subject alt name = smtp1.covertpacket.com
  > subject alt name = smtp3.covertpacket.com
  > subject alt name = smtp2.covertpacket.com
Done.

Parsing URL host header...
  > Host (url): cpftest.covertpacket.com
  > Applicable host wildcards to consider: *.covertpacket.com
Done.

ERROR - hostname does not match certificate CN or subject alt names.
  > potential security issue!


Altered target hostname to mail.covertpacket.com to accommodate wildcard
certificate and non-matching domain

<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>

CPF-TIMER> Starting perl timer
CPF-DNS>    Beginning CPF lookup on mail.covertpacket.com [primary]
```

```
CPF-DNS>    Hostname mail.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.104

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 +all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "+all"
CPF-PARSE> Qualifier: +; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     all   NULL  primary     mail.covertpacket.com
)

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  Processing ace #0:
ACL>  MATCH - Peer cert matches "all" mechanism with qualifier "+" --> Pass

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

CPF result is [pass]
```

Test 6

Purpose

Demonstrate that default qualifier "+" is applied on directives lacking an explicit qualifier

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

Assertion:

CPF Result: Pass
CPF Action: Allow
Reason: CPF should match on the SHA1 signature of the certificate and apply the default "+" qualifier.

Analysis

The + qualifier was successfully applied, as indicated by the log line "*Adding default "+" qualifier to directive*".

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.40	5.66	5.65	5.57
Time Δ (ms) - ACL Parse	0.98	1.95	1.96	1.63
Time Δ (ms) - Total	67.24	68.95	73.02	69.74
DNS bandwidth (bytes)	494	494	494	494
# of queries	2	2	2	2
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	100.33	100.23	96.73	99.10
Time Δ (ms) - ACL Parse	0.82	5.30	5.81	3.98
Time Δ (ms) - Total	183.17	175.40	173.11	177.23
DNS bandwidth (bytes)	494	494	494	494
# of queries	2	2	2	2
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>
```

```
CPF-DNS>    Beginning CPF lookup on  cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -
all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
Data = "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"

CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     hash_sha1   a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
      primary     cpftest.covertpacket.com
CPF_ACL_1: 1      -     all   NULL  primary     cpftest.covertpacket.com

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
COMPARE>    Peer Hash: a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
COMPARE>    ACL Hash:  a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
ACL>  MATCH - Peer cert and CPF record share hash
"a17e5dc70a7c31e1e6b819d450f0646fe33d7f81" with qualifier "+" --> Pass
TIMER>      Processed CPF ACLs in 0.98 ms

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

CPF result is [pass]
```

Test 7

Purpose

Demonstrate that the "-" qualifier will be enforced on the hash mechanism.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 -hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

Assertion:

CPF Result: Fail
CPF Action: Block
Reason: The certificate hash will match and the "-" will be enforced, blocking the connection.

Analysis

The browser successfully paired the server certificate hash to the appropriate directive, and as a result the connection was blocked. The log line *"MATCH - Peer cert and CPF record share hash "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81" with qualifier "-" --> Fail"* indicates that the policy was properly interpreted.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	6.56	5.69	5.79	6.01
Time Δ (ms) - ACL Parse	1.92	1.89	1.83	1.88
Time Δ (ms) - Total	81.41	68.55	78.10	76.02
DNS bandwidth (bytes)	495	495	495	495
# of queries	2	2	2	2
CPF Result	Fail	Fail	Fail	
CPF Action	Block	Block	Block	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	96.07	98.97	96.83	97.29
Time Δ (ms) - ACL Parse	5.78	1.64	6.03	4.48
Time Δ (ms) - Total	171.22	181.58	181.79	178.20
DNS bandwidth (bytes)	495	495	495	495
# of queries	2	2	2	2
CPF Result	Fail	Fail	Fail	
CPF Action	Block	Block	Block	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 -hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
-all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "-
hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
CPF-PARSE> Qualifier: -; Mechanism: hash_sha1; Data:
a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
Data = "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"

CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      -     hash_sha1   a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
      primary     cpftest.covertpacket.com
CPF_ACL_1: 1      -     all   NULL  primary     cpftest.covertpacket.com

CPF-TIMER> Finished processing Perl script in 6.56 ms (relative to start of
Perl script, not Java)

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>
<===============  ClientApp ACL Inspection STARTS HERE  ===============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
COMPARE>    Peer Hash: a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
COMPARE>    ACL Hash:  a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
ACL>  MATCH - Peer cert and CPF record share hash
"a17e5dc70a7c31e1e6b819d450f0646fe33d7f81" with qualifier "-" --> Fail

<===============  ClientApp ACL Inspection ENDS HERE  ===============>

CPF result is [fail]
```

Test 8

Purpose

Demonstrate that the "~" qualifier will be enforced on the hash mechanism.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 ~hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

Assertion:

CPF Result: SoftFail
CPF Action: Warn
Reason: The certificate hash will match and the "~" will be enforced as "SoftFail". As a result, the client application should warn the end-user regarding the untrusted certificate.

Analysis

The policy was properly enforced, as indicated by the log line "*MATCH - Peer cert and CPF record share hash "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81" with qualifier "~" --> SoftFail*" .

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.87	5.74	5.58	5.73
Time Δ (ms) - ACL Parse	1.51	1.81	0.98	1.43
Time Δ (ms) - Total	73.63	71.66	76.19	73.83
DNS bandwidth (bytes)	495	495	495	495
# of queries	2	2	2	2
CPF Result	SoftFail	SoftFail	SoftFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	91.20	96.20	96.22	94.54
Time Δ (ms) - ACL Parse	0.98	5.55	1.61	2.71
Time Δ (ms) - Total	154.30	176.49	167.85	166.21
DNS bandwidth (bytes)	495	495	495	495
# of queries	2	2	2	2
CPF Result	SoftFail	SoftFail	SoftFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 ~hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
-all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"~hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
CPF-PARSE> Qualifier: ~; Mechanism: hash_sha1; Data:
a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
Data = "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"

CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      ~     hash_sha1   a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
      primary     cpftest.covertpacket.com
CPF_ACL_1: 1      -     all   NULL  primary     cpftest.covertpacket.com

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
COMPARE>    Peer Hash: a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
COMPARE>    ACL Hash:  a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
ACL>  MATCH - Peer cert and CPF record share hash
"a17e5dc70a7c31e1e6b819d450f0646fe33d7f81" with qualifier "~" --> SoftFail

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

CPF result is [softfail]
```

Test 9

Purpose

Demonstrate that the "?" qualifier will be enforced on the hash mechanism.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 ?hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

Assertion:

CPF Result: Neutral
CPF Action: Allow
Reason: The certificate hash will match and the "?" will be enforced, permitting the connection.

Analysis

CPF processed the policy and triggered on the SHA1 signature, returning the result "neutral" as anticipated. The "neutral" result is implemented for debugging and does not imply trust or lack of trust. Thus, it is enforced just like the "none" result.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	6.44	5.76	5.85	6.02
Time Δ (ms) - ACL Parse	1.89	1.86	0.88	1.54
Time Δ (ms) - Total	71.03	69.92	78.22	73.06
DNS bandwidth (bytes)	495	495	495	495
# of queries	2	2	2	2
CPF Result	Neutral	Neutral	Neutral	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	96.14	99.27	96.06	97.16
Time Δ (ms) - ACL Parse	4.28	1.78	5.26	3.77
Time Δ (ms) - Total	173.57	173.78	173.92	173.76
DNS bandwidth (bytes)	495	495	495	495
# of queries	2	2	2	2
CPF Result	Neutral	Neutral	Neutral	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 ?hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
-all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"?hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
CPF-PARSE> Qualifier: ?; Mechanism: hash_sha1; Data:
a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
Data = "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"

CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      ?     hash_sha1   a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
      primary     cpftest.covertpacket.com
CPF_ACL_1: 1      -     all   NULL  primary     cpftest.covertpacket.com

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>
<===============  ClientApp ACL Inspection STARTS HERE  ===============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
COMPARE>    Peer Hash: a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
COMPARE>    ACL Hash:  a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
ACL>  MATCH - Peer cert and CPF record share hash
"a17e5dc70a7c31e1e6b819d450f0646fe33d7f81" with qualifier "?" --> Neutral

<===============  ClientApp ACL Inspection ENDS HERE  ===============>

CPF result is [neutral]
```

Test 10

Purpose

Demonstrate that CPF will default to "~all" (SoftFail) when directives are available but none are matched.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 hash_sha1:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"

Assertion:

CPF Result: SoftFail
CPF Action: Warn
Reason: The explicit directives will not match, and thus the default result of "SoftFail".

Analysis

The server's certificate hash does not match the explicit "aaaaa…" hash specified in the CPF record, and an "all" mechanism was not provided. Thus, the default was applied as shown in the log line *"MATCH DEFAULT - no explicit rules match, enforcing implicit rule "SoftFail""*.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.78	6.06	5.47	5.77
Time Δ (ms) - ACL Parse	2.01	0.91	1.00	1.31
Time Δ (ms) - Total	72.10	79.31	68.51	73.31
DNS bandwidth (bytes)	489	489	489	489
# of queries	2	2	2	2
CPF Result	SoftFail	SoftFail	SoftFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	99.64	97.88	98.88	98.80
Time Δ (ms) - ACL Parse	7.12	1.77	5.07	4.65
Time Δ (ms) - Total	181.45	177.91	177.50	178.95
DNS bandwidth (bytes)	489	489	489	489
# of queries	2	2	2	2
CPF Result	SoftFail	SoftFail	SoftFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 hash_sha1:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"hash_sha1:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Data = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"


CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     hash_sha1   aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
      primary     cpftest.covertpacket.com

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>
<===============  ClientApp ACL Inspection STARTS HERE  ===============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
COMPARE>    Peer Hash: a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
COMPARE>    ACL Hash:  aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
ACL>  NO MATCH
ACL>  MATCH DEFAULT - no explicit rules match, enforcing implicit rule
"SoftFail"

<===============  ClientApp ACL Inspection ENDS HERE  ===============>

CPF result is [softfail]
```

Test 11

Purpose

Demonstrate that the CPF interpreter will detect a malformed hash mechanism

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 ?hash_sha1:ada6986914e5e766f7df0539e9cac45ae -all"

Assertion:

CPF Result: PermFail
CPF Action: Warn
Reason: The hash is only 34 characters long, whereas a real SHA1 hash is 40 characters long. The CPF interpreter should detect the issue and return the "PermFail" result.

Analysis

The interpreter successfully detected the malformed entry, as shown in the log line "*CPF-PARSE> ERROR: Malformed mechanism hash data ("hash_sha1" [sha1] --> "ada6986914e5e766f7df0539e9cac45ae")*". Since a CPF record exists but cannot be interpreted, the "PermFail" result must be returned.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.79	5.95	5.81	5.85
Time Δ (ms) - ACL Parse	0.08	0.08	0.13	0.10
Time Δ (ms) - Total	77.67	77.95	69.26	74.96
DNS bandwidth (bytes)	488	488	488	488
# of queries	2	2	2	2
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	97.35	97.21	95.74	96.77
Time Δ (ms) - ACL Parse	0.65	0.58	0.59	0.61
Time Δ (ms) - Total	171.19	172.13	169.73	171.02
DNS bandwidth (bytes)	488	488	488	488
# of queries	2	2	2	2
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 ?hash_sha1:ada6986914e5e766f7df0539e9cac45ae -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "?hash_sha1:ada6986914e5e766f7df0539e9cac45ae"
CPF-PARSE> Qualifier: ?; Mechanism: hash_sha1; Data:
ada6986914e5e766f7df0539e9cac45ae
Data = "ada6986914e5e766f7df0539e9cac45ae"

CPF-PARSE> ERROR: Malformed mechanism hash data ("hash_sha1" [sha1] -->
"ada6986914e5e766f7df0539e9cac45ae")
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: PermFail
CPF_ERROR_0:      Malformed mechanism hash data ("hash_sha1" [sha1] -->
"ada6986914e5e766f7df0539e9cac45ae")

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>

CPF result is [permfail]
```

Test 12

Purpose
Demonstrate that CPF validates the protocol version

Experiment Configuration
URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com →
"hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

Assertion:
CPF Result: PermFail
CPF Action: Warn
Reason: A version is required in each CPF record per RFC.

Analysis
The protocol requires a version number to be processed properly. Since one was not provided, the record validation fails and the interpreter returns "PermFail".

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.68	5.65	6.23	5.85
Time Δ (ms) - ACL Parse	0.10	0.12	0.09	0.10
Time Δ (ms) - Total	68.80	77.12	70.82	72.25
DNS bandwidth (bytes)	490	490	490	490
# of queries	2	2	2	2
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	98.71	95.06	99.09	97.62
Time Δ (ms) - ACL Parse	0.54	0.54	0.47	0.52
Time Δ (ms) - Total	174.93	172.41	176.84	174.73
DNS bandwidth (bytes)	490	490	490	490
# of queries	2	2	2	2
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>
```

```
CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

CPF-PARSE> CPF version detected as "" (originally "")
CPF-PARSE> CPF directive #0 =
"hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
Data = "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"

CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: PermFail
CPF_ERROR_0:      No CPF version specified

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>

CPF result is [permfail]
```

Test 13

Purpose

Demonstrate that the CPF interpreter will detect multiple CPF records for a hostname and result in protocol failure.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 ?hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
cpftest.covertpacket.com → "v=1 ~all"

Assertion:

CPF Result: PermFail
CPF Action: Warn
Reason: The CPF protocol allows only one CPF record for a given hostname, per RFC. It should result in an error when multiple records are present.

Analysis

In this situation, the CPF record fails validation as indicated by the log line "*CPF_ERROR_0: CPF-DNS> Multiple CPF records published for cpftest.covertpacket.com*". When multiple DNS records are published, there is no guarantee that they will be returned to the client in the intended order. In the log, the DNS responses are actually listed in the reverse order – the "~all" is reported first, followed by the more specific "?hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81". The RFC limits CPF to one record to ensure that the policy is enforced consistently.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.49	5.74	5.71	5.65
Time Δ (ms) - ACL Parse	0.06	0.06	0.06	0.06
Time Δ (ms) - Total	66.42	66.70	66.60	66.57
DNS bandwidth (bytes)	511	511	511	511
# of queries	2	2	2	2
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	99.54	97.82	96.56	97.97
Time Δ (ms) - ACL Parse	0.32	0.22	0.23	0.26
Time Δ (ms) - Total	180.88	168.64	178.71	176.08
DNS bandwidth (bytes)	511	511	511	511
# of queries	2	2	2	2
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 ~all"
CPF-DNS>          o "v=1 ?hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"


CPF_RESULT: PermFail
CPF_ERROR_0:      CPF-DNS>    Multiple CPF records published for
cpftest.covertpacket.com.

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
CPF result is [permfail]
```

Test 14

Purpose

Demonstrate that the CPF interpreter will detect an invalid mechanism

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 #daab9b6516e5e766fc67952774e28bf7b8a7abc6"

Assertion:

CPF Result: PermFail
CPF Action: Warn
Reason: A known CPF mechanism is not provided.

Analysis

The interpreter automatically applied the "+" qualifier since no valid qualifier was specified. The "#" is treated as part of the mechanism instead. Additionally, since no valid mechanism was provided, the interpreter throws the error "*CPF_ERROR_0: Invalid CPF mechanism "#daab9b6516e5e766fc67952774e28bf7b8a7abc6" for cpftest.covertpacket.com*" and returns "PermFail".

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.72	5.60	5.71	5.68
Time Δ (ms) - ACL Parse	0.14	0.11	0.06	0.10
Time Δ (ms) - Total	75.88	67.23	66.36	69.82
DNS bandwidth (bytes)	480	480	480	480
# of queries	2	2	2	2
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	95.55	99.25	94.76	96.52
Time Δ (ms) - ACL Parse	0.43	0.62	0.46	0.50
Time Δ (ms) - Total	171.37	173.32	167.05	170.58
DNS bandwidth (bytes)	480	480	480	480
# of queries	2	2	2	2
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 #daab9b6516e5e766fc67952774e28bf7b8a7abc6"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "#daab9b6516e5e766fc67952774e28bf7b8a7abc6"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism:
#daab9b6516e5e766fc67952774e28bf7b8a7abc6; Data: primary
CPF-PARSE> Invalid CPF mechanism "#daab9b6516e5e766fc67952774e28bf7b8a7abc6"

CPF_RESULT: PermFail
CPF_ERROR_0:      Invalid CPF mechanism
"#daab9b6516e5e766fc67952774e28bf7b8a7abc6" for cpftest.covertpacket.com

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>

CPF result is [permfail]
```

Test 15

Purpose
Demonstrate that SSL connections are permitted when a CPF record is not advertised

Experiment Configuration
URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): *N/A*

Assertion:
CPF Result: None
CPF Action: Allow
Reason: No CPF record is advertised, thus there is no policy associated with the service to enforce.

Analysis
Since no CPF record exists, the "none" result is returned.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	4.05	4.14	4.06	4.08
Time Δ (ms) - ACL Parse	0.07	0.07	0.07	0.07
Time Δ (ms) - Total	64.68	65.19	65.80	65.22
DNS bandwidth (bytes)	439	439	439	439
# of queries	2	2	2	2
CPF Result	None	None	None	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	90.93	91.47	92.09	91.50
Time Δ (ms) - ACL Parse	0.26	0.30	0.31	0.29
Time Δ (ms) - Total	160.57	157.91	163.06	160.51
DNS bandwidth (bytes)	439	439	439	439
# of queries	2	2	2	2
CPF Result	None	None	None	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]
```

```
CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    No CPF record found.


CPF_RESULT: None

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>

CPF result is [none]
```

Test 16

Purpose
Demonstrate that CPF follows "include" mechanisms

Experiment Configuration
URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 include:_wcc_cpf.covertpacket.com hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"
_wcc_cpf.covertpacket.com → "v=1 hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192 -all"

Assertion:
CPF Result: Pass
CPF Action: Allow
Reason: The CPF interpreter should follow the include, not match, and then match the proceeding

Analysis
The CPF interpreter queries for the initial CPF record for "cpftest.covertpacket.com" and begins to parse each directive. When the parser reaches the "include" directive for "_wcc_cpf.covertpacket.com", it must query for the CPF record of this hostname and parse it. All directives are parsed, validated, and appended to the policy ACL except for the "all" mechanism.

Once the included record has been completely processed, the interpreter will move on to the next directive in the original process list. The final ACL appears as follows:

#	Qualifier	Mechanism	Source	Associated with Hostname
0	+	184eee13a42e63efdd08d663a03a7fb6e6244192	***Include***	_wcc_cpf.covertpacket.com
1	+	a17e5dc70a7c31e1e6b819d450f0646fe33d7f81	Primary	cpftest.covertpacket.com
2	-	ALL	Primary	cpftest.covertpacket.com

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	7.00	6.83	6.91	6.91
Time Δ (ms) - ACL Parse	1.90	2.31	2.23	2.15
Time Δ (ms) - Total	73.95	75.05	73.14	74.05
DNS bandwidth (bytes)	805	805	805	805
# of queries	3	3	3	3
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	145.37	139.44	143.95	142.92
Time Δ (ms) - ACL Parse	3.32	4.02	2.27	3.20
Time Δ (ms) - Total	225.96	227.51	221.58	225.02
DNS bandwidth (bytes)	805	805	805	805
# of queries	3	3	3	3
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_wcc_cpf.covertpacket.com
hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_wcc_cpf.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _wcc_cpf.covertpacket.com
---- START INCLUDE -----------------
CPF-DNS>    Beginning CPF lookup on _wcc_cpf.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192 -
all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
184eee13a42e63efdd08d663a03a7fb6e6244192
Data = "184eee13a42e63efdd08d663a03a7fb6e6244192"
```

```
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 =
"hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
Data = "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"

CPF-PARSE> CPF directive #2 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     hash_sha1   184eee13a42e63efdd08d663a03a7fb6e6244192
      include     _wcc_cpf.covertpacket.com
CPF_ACL_1: 1      +     hash_sha1   a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
      primary     cpftest.covertpacket.com
CPF_ACL_2: 1      -     all   NULL  primary     cpftest.covertpacket.com

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
COMPARE>    Peer Hash: a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
COMPARE>    ACL Hash:  184eee13a42e63efdd08d663a03a7fb6e6244192
ACL>  NO MATCH
ACL>  Processing ace #1:
COMPARE>    Peer Hash: a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
COMPARE>    ACL Hash:  a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
ACL>  MATCH - Peer cert and CPF record share hash
"a17e5dc70a7c31e1e6b819d450f0646fe33d7f81" with qualifier "+" --> Pass

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

CPF result is [pass]
```

Test 17

Purpose
Demonstrate that the CPF interpreter limits the number of DNS queries per RFC

Experiment Configuration
URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 include:_loop.covertpacket.com -all"
_loop.covertpacket.com → "v=1 include:_loop.covertpacket.com"

Assertion:
CPF Result: PermFail
CPF Action: Warn
Reason: The CPF records will result in a loop, which should be suspended after the tenth CPF query.

Analysis
A maximum of ten (10) queries are permitted to resolve the full CPF information for a hostname. If this threshold is exceeded, the interpreter should treat the condition as a protocol error, return the "PermFail" result, and exit. In this test, the "include" includes itself, creating an infinite loop. The query count restriction breaks the loop after ten attempts, as demonstrated by the log line "*CPF_ERROR_0: Too many DNS lookups performed for CPF information (max = 10)*". Note that the total number of queries equals eleven for each instance of this test because the initial A-record lookup fpr cpftest.covertpacket.com is counted in the final tally, but not in the ten-query CPF limit.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	21.88	22.10	23.50	22.49
Time Δ (ms) - ACL Parse	0.06	0.06	0.06	0.06
Time Δ (ms) - Total	89.83	86.51	89.80	88.71
DNS bandwidth (bytes)	2733	2733	2733	2733
# of queries	11	11	11	11
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	513.65	495.21	505.14	504.67
Time Δ (ms) - ACL Parse	0.20	0.21	0.06	0.16
Time Δ (ms) - Total	580.24	577.08	578.20	578.51
DNS bandwidth (bytes)	2733	2733	2733	2733
# of queries	11	11	11	11
CPF Result	PermFail	PermFail	PermFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
```

```
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
```

```
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_loop.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_loop.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _loop.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _loop.covertpacket.com [include]

CPF-DNS>    Too many DNS lookups performed for CPF information (max = 10).

---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
```

```
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: PermFail
CPF_ERROR_0:      Too many DNS lookups performed for CPF information (max =
10).

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>

CPF result is [permfail]
```

Test 18

Purpose

Demonstrate that CPF resolver will conduct lookups based on the URL hostname when the service is identified by a valid wildcard certificate.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: *.covertpacket.com (CN)
Certificate Hash: (sha1) 184eee13a42e63efdd08d663a03a7fb6e6244192
CPF records text (DNS): *N/A*

Assertion:

CPF Result: None
CPF Action: Allow
Reason: The CPF resolver will query for CPF records associated with cpftest.covertpacket.com, which will yield no results.

Analysis

The CPF lookup was conducted for cpftest.covertpacket.com, as indicated by the log line "*Beginning CPF lookup on cpftest.covertpacket.com*".

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	3.88	4.11	4.19	4.06
Time Δ (ms) - ACL Parse	0.06	0.07	0.07	0.07
Time Δ (ms) - Total	63.23	64.49	65.09	64.27
DNS bandwidth (bytes)	439	439	439	439
# of queries	2	2	2	2
CPF Result	None	None	None	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	89.03	92.55	90.83	90.80
Time Δ (ms) - ACL Parse	0.24	0.32	0.24	0.27
Time Δ (ms) - Total	161.85	163.57	158.01	161.14
DNS bandwidth (bytes)	439	439	439	439
# of queries	2	2	2	2
CPF Result	None	None	None	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
INIT - Host and Cert analysis
```

```
URL = https://cpftest.covertpacket.com
domain = covertpacket.com
hostname = cpftest.covertpacket.com
cpftest.covertpacket.com resolves to 192.168.1.102

Retrieving certificate attributes...
  > common name = *.covertpacket.com
  > subject alt name = <None>
Done.


Parsing URL host header...
  > Host (url): cpftest.covertpacket.com
  > Applicable host wildcards to consider: *.covertpacket.com
Done.

Cert hostname review PASSED - valid for cpftest.covertpacket.com

<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.102

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    No CPF record found.


CPF_RESULT: None

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>

CPF result is [none]
```

Test 19

Purpose

Demonstrate that an "include" can be used to link a host's CPF information to a corresponding wildcard certificate.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: *.covertpacket.com (CN)
Certificate Hash: (sha1) 184eee13a42e63efdd08d663a03a7fb6e6244192

CPF records text (DNS): cpftest.covertpacket.com → "v=1 include:_wcc_cpf.covertpacket.com -all"
_wcc_cpf.covertpacket.com → v=1
hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192 -all

Assertion:

CPF Result: Pass
CPF Action: Allow
Reason: The CPF interpreter should trigger on the hash mechanism noted in the wildcard certificate CPF record, which is included from the host's CPF record.

Analysis

The CPF interpreter follows the CPF include and inherits the directive "hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192", which matches the wildcard certificate SHA1 hash. As a result, the connection is permitted.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	6.86	6.85	6.88	6.86
Time Δ (ms) - ACL Parse	1.01	1.93	1.87	1.60
Time Δ (ms) - Total	73.05	69.63	72.85	71.84
DNS bandwidth (bytes)	754	754	754	754
# of queries	3	3	3	3
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	140.78	145.52	138.50	141.60
Time Δ (ms) - ACL Parse	1.71	5.37	1.99	3.02
Time Δ (ms) - Total	221.77	233.50	219.66	224.98
DNS bandwidth (bytes)	754	754	754	754
# of queries	3	3	3	3
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on  cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.102

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:_wcc_cpf.covertpacket.com -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:_wcc_cpf.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: _wcc_cpf.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on _wcc_cpf.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192 -
all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
184eee13a42e63efdd08d663a03a7fb6e6244192
Data = "184eee13a42e63efdd08d663a03a7fb6e6244192"
```

```
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     hash_sha1   184eee13a42e63efdd08d663a03a7fb6e6244192
      include     _wcc_cpf.covertpacket.com
CPF_ACL_1: 1      -     all   NULL  primary     cpftest.covertpacket.com

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "184eee13a42e63efdd08d663a03a7fb6e6244192"
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  184eee13a42e63efdd08d663a03a7fb6e6244192
ACL>  MATCH - Peer cert and CPF record share hash
"184eee13a42e63efdd08d663a03a7fb6e6244192" with qualifier "+" --> Pass

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

CPF result is [pass]
```

Test 20

Purpose

Demonstrate that large CPF records (>512 bytes in size) will be resolvable via DNS

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: *.covertpacket.com (CN)
Certificate Hash: (sha1) 184eee13a42e63efdd08d663a03a7fb6e6244192

CPF records text (DNS): cpftest.covertpacket.com → "v=1
hash_sha1:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
hash_sha1:bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
hash_sha1:ccccccccccccccccccccccccccccccccccccccccc "
" hash_sha1:ddddddddddddddddddddddddddddddddddddddddd
hash_sha1:eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
hash_sha1:ffffffffffffffffffffffffffffffffffffffff "
" hash_sha1:1111111111111111111111111111111111111111
hash_sha1:22222222222222222222222222222222222222222
hash_sha1:333333333333333333333333333333333333333333 "
" hash_sha1:44444444444444444444444444444444444444444
hash_sha1:55555555555555555555555555555555555555555
hash_sha1:66666666666666666666666666666666666666666 "
"hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192 -all"

Assertion:

CPF Result: Pass
CPF Action: Allow
Reason: The CPF record should be retrieved via DNS using TCP, and will match the final SHA1 hash mechanism.

Analysis

As shown in the DNS packet capture summary, the DNS TXT response in frame 25 has the truncate flag set. This tells the client application that the full response is too large for a typical DNS UDP packet, and should renegotiate via TCP. Frames 26 - 28 show the TCP handshake, followed by the request and 833 byte response in frame 31 (highlighted yellow). This functionality is built into the DNS protocol, and thus completely transparent to CPF.

Once the full CPF record has been acquired, the interpreter parses each directive and builds the ACL. The certificate hash matches the last entry in the ACL and the connection is permitted.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	7.19	6.96	7.16	7.10
Time Δ (ms) - ACL Parse	12.72	5.20	5.08	7.67
Time Δ (ms) - Total	83.30	75.94	76.65	78.63
DNS bandwidth (bytes)	1330	1330	1330	1330
# of queries	3	3	3	3
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	181.83	180.92	184.43	182.39
Time Δ (ms) - ACL Parse	3.01	4.02	3.92	3.65
Time Δ (ms) - Total	258.97	265.41	270.01	264.80
DNS bandwidth (bytes)	1330	1330	1330	1330
# of queries	3	3	3	3
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on  cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.102

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 hash_sha1:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
hash_sha1:bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
hash_sha1:cccccccccccccccccccccccccccccccccccccccc
hash_sha1:dddddddddddddddddddddddddddddddddddddddd
hash_sha1:eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
hash_sha1:ffffffffffffffffffffffffffffffffffffffff
hash_sha1:1111111111111111111111111111111111111111
hash_sha1:2222222222222222222222222222222222222222
hash_sha1:3333333333333333333333333333333333333333
hash_sha1:4444444444444444444444444444444444444444
hash_sha1:5555555555555555555555555555555555555555
hash_sha1:6666666666666666666666666666666666666666
hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192 -all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"hash_sha1:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Data = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
```

```
CPF-PARSE> CPF directive #1 =
"hash_sha1:bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
Data = "bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb"

CPF-PARSE> CPF directive #2 =
"hash_sha1:ccccccccccccccccccccccccccccccccccccccccc"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
ccccccccccccccccccccccccccccccccccccccccc
Data = "ccccccccccccccccccccccccccccccccccccccccc"

CPF-PARSE> CPF directive #3 =
"hash_sha1:ddddddddddddddddddddddddddddddddddddddddd"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
ddddddddddddddddddddddddddddddddddddddddd
Data = "ddddddddddddddddddddddddddddddddddddddddd"

CPF-PARSE> CPF directive #4 =
"hash_sha1:eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
Data = "eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee"

CPF-PARSE> CPF directive #5 =
"hash_sha1:ffffffffffffffffffffffffffffffffffffffff"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
ffffffffffffffffffffffffffffffffffffffff
Data = "ffffffffffffffffffffffffffffffffffffffff"

CPF-PARSE> CPF directive #6 =
"hash_sha1:1111111111111111111111111111111111111111"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
1111111111111111111111111111111111111111
Data = "1111111111111111111111111111111111111111"

CPF-PARSE> CPF directive #7 =
"hash_sha1:2222222222222222222222222222222222222222"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
2222222222222222222222222222222222222222
Data = "2222222222222222222222222222222222222222"

CPF-PARSE> CPF directive #8 =
"hash_sha1:3333333333333333333333333333333333333333"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
3333333333333333333333333333333333333333
Data = "3333333333333333333333333333333333333333"

CPF-PARSE> CPF directive #9 =
"hash_sha1:4444444444444444444444444444444444444444"
CPF-PARSE> Adding default "+" qualifier to directive
```

```
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
4444444444444444444444444444444444444444
Data = "4444444444444444444444444444444444444444"

CPF-PARSE> CPF directive #10 =
"hash_sha1:5555555555555555555555555555555555555555"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
5555555555555555555555555555555555555555
Data = "5555555555555555555555555555555555555555"

CPF-PARSE> CPF directive #11 =
"hash_sha1:6666666666666666666666666666666666666666"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
6666666666666666666666666666666666666666
Data = "6666666666666666666666666666666666666666"

CPF-PARSE> CPF directive #12 =
"hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
184eee13a42e63efdd08d663a03a7fb6e6244192
Data = "184eee13a42e63efdd08d663a03a7fb6e6244192"

CPF-PARSE> CPF directive #13 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     hash_sha1   aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
      primary     cpftest.covertpacket.com
CPF_ACL_1: 1      +     hash_sha1   bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
      primary     cpftest.covertpacket.com
CPF_ACL_2: 1      +     hash_sha1   ccccccccccccccccccccccccccccccccccccccccc
      primary     cpftest.covertpacket.com
CPF_ACL_3: 1      +     hash_sha1   ddddddddddddddddddddddddddddddddddddddddd
      primary     cpftest.covertpacket.com
CPF_ACL_4: 1      +     hash_sha1   eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
      primary     cpftest.covertpacket.com
CPF_ACL_5: 1      +     hash_sha1   ffffffffffffffffffffffffffffffffffffffff
      primary     cpftest.covertpacket.com
CPF_ACL_6: 1      +     hash_sha1   1111111111111111111111111111111111111111
      primary     cpftest.covertpacket.com
CPF_ACL_7: 1      +     hash_sha1   2222222222222222222222222222222222222222
      primary     cpftest.covertpacket.com
CPF_ACL_8: 1      +     hash_sha1   3333333333333333333333333333333333333333
      primary     cpftest.covertpacket.com
CPF_ACL_9: 1      +     hash_sha1   4444444444444444444444444444444444444444
      primary     cpftest.covertpacket.com
CPF_ACL_10: 1     +     hash_sha1   5555555555555555555555555555555555555555
      primary     cpftest.covertpacket.com
CPF_ACL_11: 1     +     hash_sha1   6666666666666666666666666666666666666666
      primary     cpftest.covertpacket.com
CPF_ACL_12: 1     +     hash_sha1   184eee13a42e63efdd08d663a03a7fb6e6244192
      primary     cpftest.covertpacket.com
CPF_ACL_13: 1     -     all   NULL  primary     cpftest.covertpacket.com
```

```
<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>
<===============  ClientApp ACL Inspection STARTS HERE  ===============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "184eee13a42e63efdd08d663a03a7fb6e6244192"
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
ACL>  NO MATCH
ACL>  Processing ace #1:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
ACL>  NO MATCH
ACL>  Processing ace #2:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  cccccccccccccccccccccccccccccccccccccccc
ACL>  NO MATCH
ACL>  Processing ace #3:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  dddddddddddddddddddddddddddddddddddddddd
ACL>  NO MATCH
ACL>  Processing ace #4:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
ACL>  NO MATCH
ACL>  Processing ace #5:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  ffffffffffffffffffffffffffffffffffffffff
ACL>  NO MATCH
ACL>  Processing ace #6:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  1111111111111111111111111111111111111111
ACL>  NO MATCH
ACL>  Processing ace #7:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  2222222222222222222222222222222222222222
ACL>  NO MATCH
ACL>  Processing ace #8:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  3333333333333333333333333333333333333333
ACL>  NO MATCH
ACL>  Processing ace #9:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  4444444444444444444444444444444444444444
ACL>  NO MATCH
ACL>  Processing ace #10:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  5555555555555555555555555555555555555555
ACL>  NO MATCH
ACL>  Processing ace #11:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  6666666666666666666666666666666666666666
ACL>  NO MATCH
ACL>  Processing ace #12:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  184eee13a42e63efdd08d663a03a7fb6e6244192
ACL>  MATCH - Peer cert and CPF record share hash
"184eee13a42e63efdd08d663a03a7fb6e6244192" with qualifier "+" --> Pass
```

<============== ClientApp ACL Inspection ENDS HERE ==============>

CPF result is [pass]

DNS Packet Capture Summary:

Frame #	Source IP	Destination IP	Dest Port	Info	Size (bytes)
22	192.168.1.40	192.168.1.144	53	Standard query A cpftest.covertpacket.com	84
23	192.168.1.144	192.168.1.40	55774	Standard query response A 192.168.1.102	135
24	192.168.1.40	192.168.1.144	53	Standard query TXT cpftest.covertpacket.com	84
25	192.168.1.144	192.168.1.40	36888	Standard query response	84
26	192.168.1.40	192.168.1.144	53	39153 > domain [SYN]	74
27	192.168.1.144	192.168.1.40	39153	domain > 39153 [SYN ACK]	74
28	192.168.1.40	192.168.1.144	53	39153 > domain [ACK]	66
29	192.168.1.40	192.168.1.144	53	Standard query TXT cpftest.covertpacket.com	110
30	192.168.1.144	192.168.1.40	39153	domain > 39153 [ACK]	66
31	192.168.1.144	192.168.1.40	39153	Standard query response TXT	833
32	192.168.1.40	192.168.1.144	53	39153 > domain [ACK]	66
33	192.168.1.40	192.168.1.144	53	39153 > domain [FIN ACK]	66
34	192.168.1.144	192.168.1.40	39153	domain > 39153 [FIN ACK]	66
35	192.168.1.40	192.168.1.144	53	39153 > domain [ACK]	66

Frame 25: 84 bytes on wire (672 bits), 84 bytes captured (672 bits)
Internet Protocol, Src: 192.168.1.144 (192.168.1.144), Dst: 192.168.1.40 (192.168.1.40)
User Datagram Protocol, Src Port: domain (53), Dst Port: 36888 (36888)
Domain Name System (response)
  [Request In: 24]
  [Time: 0.000393000 seconds]
  Transaction ID: 0x8be6
  Flags: 0x8680 (Standard query response, No error)
    1... .... = Response: Message is a response
    .000 0... .... = Opcode: Standard query (0)
    .... .1.. .... = Authoritative: Server is an authority for domain
    .... ..1. .... = Truncated: Message is truncated
    ....0 .... = Recursion desired: Don't do query recursively
    .... 1... .... = Recursion available: Server can do recursive queries
    .... .0.. .... = Z: reserved (0)
    ....0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
    ....0 .... = Non-authenticated data: Unacceptable
    .... 0000 = Reply code: No error (0)
  Questions: 1
  Answer RRs: 0

Test 21

Purpose

Demonstrate that CPF resolver will conduct lookups based on the URL hostname when the service is identified by a certificate subject-alternative-name (SAN).

Experiment Configuration

URL: https://smtp1.covertpacket.com

Certificate Identities: mail.covertpacket.com (CN)
smtp1.covertpacket.com (SAN)
smtp2.covertpacket.com (SAN)
smtp3.covertpacket.com (SAN)
smtp4.covertpacket.com (SAN)

Certificate Hash: (sha1) cce9cdb442743559bc8116981af6c339e2c772bb

CPF records text (DNS): smtp1.covertpacket.com → "v=1 include:mail.covertpacket.com ~all"
mail.covertpacket.com → "v=1 hash_sha1:cce9cdb442743559bc8116981af6c339e2c772bb -all"

Assertion:

CPF Result: Pass
CPF Action: Allow
Reason: The lookup will be conducted against smtp1.covertpacket.com, which includes the CPF record for mail.covertpacket.com. The included record permits the use of this hash.

Analysis

The lookup was conducted for smtp1.covertpacket.com since it is a valid subject-alternative-name for the certificate. The policy for this hostname permits the certificate hash for the service, thus the connection is permitted.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	6.80	6.97	6.63	6.80
Time Δ (ms) - ACL Parse	1.44	1.72	1.84	1.67
Time Δ (ms) - Total	74.56	72.86	69.09	72.17
DNS bandwidth (bytes)	734	734	734	734
# of queries	3	3	3	3
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	142.04	144.22	140.69	142.32
Time Δ (ms) - ACL Parse	2.12	1.58	1.87	1.86
Time Δ (ms) - Total	223.22	231.05	221.29	225.19
DNS bandwidth (bytes)	734	734	734	734
# of queries	3	3	3	3
CPF Result	Pass	Pass	Pass	
CPF Action	Allow	Allow	Allow	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on smtp1.covertpacket.com [primary]

CPF-DNS>    Hostname smtp1.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.104

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 include:mail.covertpacket.com ~all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "include:mail.covertpacket.com"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: include; Data: mail.covertpacket.com
---- START INCLUDE ------------------
CPF-DNS>    Beginning CPF lookup on mail.covertpacket.com [include]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 hash_sha1:cce9cdb442743559bc8116981af6c339e2c772bb -
all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"hash_sha1:cce9cdb442743559bc8116981af6c339e2c772bb"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
cce9cdb442743559bc8116981af6c339e2c772bb
Data = "cce9cdb442743559bc8116981af6c339e2c772bb"
```

```
CPF-PARSE> CPF directive #1 = "-all"
CPF-PARSE> Qualifier: -; Mechanism: all; Data: include
CPF-PARSE> Discarding "all" mechanism (mode=include)
---- END INCLUDE   ------------------
CPF-PARSE> CPF directive #1 = "~all"
CPF-PARSE> Qualifier: ~; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     hash_sha1   cce9cdb442743559bc8116981af6c339e2c772bb
      include     mail.covertpacket.com
CPF_ACL_1: 1      ~     all   NULL  primary     smtp1.covertpacket.com

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "cce9cdb442743559bc8116981af6c339e2c772bb"
COMPARE>    Peer Hash: cce9cdb442743559bc8116981af6c339e2c772bb
COMPARE>    ACL Hash:  cce9cdb442743559bc8116981af6c339e2c772bb
ACL>  MATCH - Peer cert and CPF record share hash
"cce9cdb442743559bc8116981af6c339e2c772bb" with qualifier "+" --> Pass

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

CPF result is [pass]
```

Test 22

Purpose

Demonstrate that CPF resolver will conduct lookups based on the certificate common name when a hostname mismatch occurs between the URL hostname and the certificate.

Experiment Configuration

URL: https://badguy.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
CPF records text (DNS): cpftest.covertpacket.com → "v=1 -hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 ~all"

Assertion:

CPF Result: Fail
CPF Action: Block
Reason: The CPF resolver should query for CPF record of cpftest.covertpacket.com, which specifies a "-" qualifier for the associated certificate signature.

Analysis

The server certificate is only valid for "cpftest.covertpacket.com", which does not match the URL hostname "badguy.covertpacket.com". A CPF lookup is conducted for "cpftest.covertpacket.com" and matches a directive that specifies the certificate hash with a "-" qualifier. As a result, the connection is blocked. This functionality is useful for blacklisting a compromised key pair.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	5.56	5.72	5.70	5.66
Time Δ (ms) - ACL Parse	1.01	1.80	2.08	1.63
Time Δ (ms) - Total	76.07	71.80	71.80	73.22
DNS bandwidth (bytes)	495	495	495	495
# of queries	2	2	2	2
CPF Result	Fail	Fail	Fail	
CPF Action	Block	Block	Block	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	96.25	97.51	95.13	96.30
Time Δ (ms) - ACL Parse	4.66	1.66	1.75	2.69
Time Δ (ms) - Total	180.06	173.98	168.43	174.16
DNS bandwidth (bytes)	495	495	495	495
# of queries	2	2	2	2
CPF Result	Fail	Fail	Fail	
CPF Action	Block	Block	Block	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

INIT - Host and Cert analysis

URL = https://badguy.covertpacket.com
domain = covertpacket.com
hostname = badguy.covertpacket.com
badguy.covertpacket.com resolves to 192.168.1.105

Retrieving certificate attributes...
  > common name = cpftest.covertpacket.com
  > subject alt name = <None>
Done.


Parsing URL host header...
  > Host (url): badguy.covertpacket.com
  > Applicable host wildcards to consider: *.covertpacket.com
Done.

ERROR - hostname does not match certificate CN or subject alt names.
  > potential security issue!

Altered target hostname to cpftest.covertpacket.com to accommodate wildcard
certificate and non-matching domain

<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    Hostname cpftest.covertpacket.com resolves to:
CPF-DNS>          o 192.168.1.101

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 -hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
~all"

CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 = "-
hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
CPF-PARSE> Qualifier: -; Mechanism: hash_sha1; Data:
a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
Data = "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"

CPF-PARSE> CPF directive #1 = "~all"
CPF-PARSE> Qualifier: ~; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      -     hash_sha1   a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
      primary     cpftest.covertpacket.com
CPF_ACL_1: 1      ~     all   NULL  primary     cpftest.covertpacket.com

<=============  CPF Query/Parsing (Perl) ENDS HERE  =============>
<==============  ClientApp ACL Inspection STARTS HERE  ==============>
```

```
ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "a17e5dc70a7c31e1e6b819d450f0646fe33d7f81"
COMPARE>    Peer Hash: a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
COMPARE>    ACL Hash:  a17e5dc70a7c31e1e6b819d450f0646fe33d7f81
ACL>  MATCH - Peer cert and CPF record share hash
"a17e5dc70a7c31e1e6b819d450f0646fe33d7f81" with qualifier "-" --> Fail

<==============  ClientApp ACL Inspection ENDS HERE  ==============>

CPF result is [fail]
```

Test 23

Purpose

Demonstrate that the CPF resolver checks for a corresponding A-record before conducting a CPF lookup.

Experiment Configuration

URL: https://badguy.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com→ "v=1 -hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

Assertion:

CPF Result: TempFail
CPF Action: Warn
Reason: The CPF resolver will not resolve an A-record for cpftest.covertpacket.com, which is required per RFC.

Analysis

The CPF resolver always verifies that an A-record exists for a hostname before querying the CPF record. In this case, an A-record does not exist for the hostname specified in the certificate. As a result, the resolver immediately returns "tempfail".

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	2.09	2.12	2.04	2.08
Time Δ (ms) - ACL Parse	0.06	0.06	0.07	0.06
Time Δ (ms) - Total	64.59	63.68	72.07	66.78
DNS bandwidth (bytes)	220	220	220	220
# of queries	1	1	1	1
CPF Result	TempFail	TempFail	TempFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	43.61	42.95	44.11	43.56
Time Δ (ms) - ACL Parse	0.25	0.24	0.24	0.24
Time Δ (ms) - Total	117.94	112.59	112.13	114.22
DNS bandwidth (bytes)	220	220	220	220
# of queries	1	1	1	1
CPF Result	TempFail	TempFail	TempFail	
CPF Action	Warn	Warn	Warn	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
INIT - Host and Cert analysis

URL = https://badguy.covertpacket.com
domain = covertpacket.com
hostname = badguy.covertpacket.com
badguy.covertpacket.com resolves to 192.168.1.105

Retrieving certificate attributes...
  > common name = cpftest.covertpacket.com
  > subject alt name = <None>
Done.


Parsing URL host header...
  > Host (url): badguy.covertpacket.com
  > Applicable host wildcards to consider: *.covertpacket.com
Done.

ERROR - hostname does not match certificate CN or subject alt names.
  > potential security issue!

Altered target hostname to cpftest.covertpacket.com to accommodate wildcard
certificate and non-matching domain

<==============  CPF Query/Parsing (Perl) STARTS HERE   ==============>

CPF-DNS>    Beginning CPF lookup on cpftest.covertpacket.com [primary]

CPF-DNS>    no A-records found for cpftest.covertpacket.com

CPF_RESULT: TempFail

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>

CPF result is [tempfail]
```

Test 24

Purpose

Demonstrate that the CPF resolver will conduct lookups based on the wildcard certificate's domain when the remote service's hostname is not valid for a wildcard certificate.

Experiment Configuration

URL: https://www.randomdomain.com

Certificate Identities: *.covertpacket.com (CN)
Certificate Hash: (sha1) 184eee13a42e63efdd08d663a03a7fb6e6244192
CPF records text (DNS): _wcc_cpf.covertpacket.com → "v=1 hash_sha1:ada6986914e5e766f7df0539e9cac45ae7f8210b -hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192 ~all"

Assertion:

CPF Result: Fail
CPF Action: Block
Reason: The CPF resolver will conduct the CPF lookup for _wcc_cpf.covertpacket.com and the connection will be blocked.

Analysis

The hostname "www.randomdomain.com" does not match the certificate for *.covertpacket.com, which requires the CPF resolver to conduct the lookup for the certificate common name. Since the server provided a wildcard certificate, the "*" is replaced with the reserved hostname "_wcc_cpf" as noted in the log "Altered target hostname to _wcc_cpf.covertpacket.com to accommodate wildcard certificate and non-matching domain". A CPF record exists for _wcc_cpf.covertpacket.com, and the resultant policy indicates that connections secured by this certificate should be blocked.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	3.89	3.87	3.52	3.76
Time Δ (ms) - ACL Parse	1.97	1.95	1.10	1.67
Time Δ (ms) - Total	67.90	69.66	71.47	69.68
DNS bandwidth (bytes)	329	329	329	329
# of queries	1	1	1	1
CPF Result	Fail	Fail	Fail	
CPF Action	Block	Block	Block	
Assertion Correct	Yes	Yes	Yes	

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	48.82	51.45	49.09	49.79
Time Δ (ms) - ACL Parse	2.66	2.12	2.47	2.42
Time Δ (ms) - Total	126.08	131.09	129.09	128.75
DNS bandwidth (bytes)	329	329	329	329
# of queries	1	1	1	1
CPF Result	Fail	Fail	Fail	
CPF Action	Block	Block	Block	
Assertion Correct	Yes	Yes	Yes	

Related log data:

```
INIT - Host and Cert analysis

URL = https://www.randomdomain.com
domain = randomdomain.com
hostname = www.randomdomain.com
www.randomdomain.com resolves to 192.168.1.106

Retrieving certificate attributes...
  > common name = *.covertpacket.com
  > subject alt name = <None>
Done.

Parsing URL host header...
  > Host (url): www.randomdomain.com
  > Applicable host wildcards to consider: *.randomdomain.com
Done.

ERROR - hostname does not match certificate CN or subject alt names.
  > potential security issue!

Altered target hostname to _wcc_cpf.covertpacket.com to accommodate wildcard
certificate and non-matching domain

<=============  CPF Query/Parsing (Perl) STARTS HERE   =============>

CPF-DNS>    Beginning CPF lookup on _wcc_cpf.covertpacket.com [primary]

CPF-DNS>    Starting lookup of CPF record...
CPF-DNS>    CPF record found:
CPF-DNS>          o "v=1 hash_sha1:ada6986914e5e766f7df0539e9cac45ae7f8210b -
hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192 ~all"
```

```
CPF-PARSE> CPF version detected as "1" (originally "v=1")
CPF-PARSE> CPF directive #0 =
"hash_sha1:ada6986914e5e766f7df0539e9cac45ae7f8210b"
CPF-PARSE> Adding default "+" qualifier to directive
CPF-PARSE> Qualifier: +; Mechanism: hash_sha1; Data:
ada6986914e5e766f7df0539e9cac45ae7f8210b
Data = "ada6986914e5e766f7df0539e9cac45ae7f8210b"

CPF-PARSE> CPF directive #1 = "-
hash_sha1:184eee13a42e63efdd08d663a03a7fb6e6244192"
CPF-PARSE> Qualifier: -; Mechanism: hash_sha1; Data:
184eee13a42e63efdd08d663a03a7fb6e6244192
Data = "184eee13a42e63efdd08d663a03a7fb6e6244192"

CPF-PARSE> CPF directive #2 = "~all"
CPF-PARSE> Qualifier: ~; Mechanism: all; Data: primary

CPF_RESULT: NULL
CPF_VERSIONS:     1
CPF_VERSION_REQUIRED:   1
CPF_ACL_0: 1      +     hash_sha1   ada6986914e5e766f7df0539e9cac45ae7f8210b
      primary     _wcc_cpf.covertpacket.com
CPF_ACL_1: 1      -     hash_sha1   184eee13a42e63efdd08d663a03a7fb6e6244192
      primary     _wcc_cpf.covertpacket.com
CPF_ACL_2: 1      ~     all   NULL  primary     _wcc_cpf.covertpacket.com

<==============  CPF Query/Parsing (Perl) ENDS HERE  ==============>
<===============  ClientApp ACL Inspection STARTS HERE  ===============>

ACL>  Processing ace #0:
ACL>  New sha1 hash needed for peer's cert pem
ACL>  Generated hash as "184eee13a42e63efdd08d663a03a7fb6e6244192"
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  ada6986914e5e766f7df0539e9cac45ae7f8210b
ACL>  NO MATCH
ACL>  Processing ace #1:
COMPARE>    Peer Hash: 184eee13a42e63efdd08d663a03a7fb6e6244192
COMPARE>    ACL Hash:  184eee13a42e63efdd08d663a03a7fb6e6244192
ACL>  MATCH - Peer cert and CPF record share hash
"184eee13a42e63efdd08d663a03a7fb6e6244192" with qualifier "-" --> Fail

<===============  ClientApp ACL Inspection ENDS HERE  ===============>

CPF result is [fail]
```

Test 7 with DNSSEC

Purpose

Investigate how CPF queries are impacted when using DNSSEC.

Experiment Configuration

URL: https://cpftest.covertpacket.com

Certificate Identities: cpftest.covertpacket.com (CN)
Certificate Hash: (sha1) a17e5dc70a7c31e1e6b819d450f0646fe33d7f81

CPF records text (DNS): cpftest.covertpacket.com → "v=1 -hash_sha1:a17e5dc70a7c31e1e6b819d450f0646fe33d7f81 -all"

Analysis

The total DNS packet data is 1211 bytes, as opposed to 495 bytes in the original version of test 7 that did not use DNSSEC. The DNSSEC "RRSIG" data for both the nameserver (NS) and target resource record are significant contributors to the increase in data. The overhead from each of the two RRSIGs is:

- 18 bytes for header information
- 36 bytes for "covertpacket.com" – *variable length based on domain name*
- 128 bytes for hashed resource record signed with 1024-bit key - *variable length based on key size*

The length of the authoritative nameserver's key pair will have a direct impact on the data that must be exchanged, as well as the domain name (to a lesser extent). The number of queries did not change when using DNSSEC, since the signed data is included in the same packet as the cleartext response.

Data Summary

LAN-based Test

Data Label	Trial 1	Trial 2	Trial 3	Average
Time Δ (ms) – CPF Query/Parse	11.58	11.91	11.62	11.70
Time Δ (ms) - ACL Parse	2.06	2.09	1.86	2.00
Time Δ (ms) - Total	123.40	123.31	122.43	123.05
DNS bandwidth (bytes)	1211	1211	1211	1211.00
# of queries	2	2	2	2.00

Internet Simulation Test

Data Label	Trial 4	Trial 5	Trial 6	Average
Time Δ (ms) – DNS CPF query	110.30	106.01	99.91	105.41
Time Δ (ms) - ACL Parse	0.92	1.95	1.71	1.53
Time Δ (ms) - Total	237.82	233.51	228.63	233.32
DNS bandwidth (bytes)	1211	1211	1211	1211.00
# of queries	2	2	2	2.00

DNS Packet Capture Summary:

Frame #	Source IP	Destination IP	Dest Port	Info	Size (bytes)
24	192.168.1.40	192.168.1.145	53	Standard query A	95

				cpftest.covertpacket.com	
25	192.168.1.145	192.168.1.40	33151	Standard query response A 192.168.1.101 RRSIG	482
26	192.168.1.40	192.168.1.145	53	Standard query TXT cpftest.covertpacket.com	95
27	192.168.1.145	192.168.1.40	53458	Standard query response TXT RRSIG	539

Frame 24: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)
Internet Protocol, Src: 192.168.1.40 (192.168.1.40), Dst: 192.168.1.145 (192.168.1.145)
User Datagram Protocol, Src Port: 33151 (33151), Dst Port: domain (53)
Domain Name System (query)
  [Response In: 25]
  Transaction ID: 0x2bc6
  Flags: 0x0120 (Standard query)
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 1
  Queries
  Additional records

Frame 25: 482 bytes on wire (3856 bits), 482 bytes captured (3856 bits)
Internet Protocol, Src: 192.168.1.145 (192.168.1.145), Dst: 192.168.1.40 (192.168.1.40)
User Datagram Protocol, Src Port: domain (53), Dst Port: 33151 (33151)
Domain Name System (response)
  [Request In: 24]
  [Time: 0.004235000 seconds]
  Transaction ID: 0x2bc6
  Flags: 0x81a0 (Standard query response, No error)
  Questions: 1
  Answer RRs: 2
  Authority RRs: 2
  Additional RRs: 1
  Queries
  Answers
  Authoritative nameservers
  Additional records

Frame 26: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)
Internet Protocol, Src: 192.168.1.40 (192.168.1.40), Dst: 192.168.1.145 (192.168.1.145)
User Datagram Protocol, Src Port: 53458 (53458), Dst Port: domain (53)
Domain Name System (query)
  [Response In: 27]
  Transaction ID: 0xf60a
  Flags: 0x0120 (Standard query)
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0

Additional RRs: 1
Queries
Additional records

Frame 27: 539 bytes on wire (4312 bits), 539 bytes captured (4312 bits)
Internet Protocol, Src: 192.168.1.145 (192.168.1.145), Dst: 192.168.1.40 (192.168.1.40)
User Datagram Protocol, Src Port: domain (53), Dst Port: 53458 (53458)
Domain Name System (response)
[Request In: 26]
[Time: 0.001008000 seconds]
Transaction ID: 0xf60a
Flags: 0x81a0 (Standard query response, No error)
Questions: 1
Answer RRs: 2
Authority RRs: 2
Additional RRs: 1
Queries
Answers
Authoritative nameservers
Additional records