Internet Protocol version 6 (IPv6)
Firewall implementation – Access Control Lists

1. *IPv6 firewall filter*

IPv6 firewall filter was created based on the IPv4 firewall filter except all the blocks are for only the ports and protocols. Currently there is no request to block any IPv6 addresses probably because NIDS haven't seen any suspicious activity with the v6 traffic. This filter is only applied on the external gateway interfaces, which are the DRENv6 test bed access points at San Diego, ARL-APG and ERDC. The initial set of IPv6 firewall filter was created by copying the blocked ports and protocols from the IPv4 firewall filter.

2. *Maintenance and updates of the IPv6 filter*

The policy is that any request from HPC-CERT or HPCMPMO to block ports or protocols for the IPv4 filter is duplicated on the IPv6 filter unless there are specific exceptions. The initial set of IPv6 firewall filter was created by copying the blocked ports and protocols from the IPv4 firewall filter.

3. *IPv6 Firewall filter example*

Attached is the actual IPv6 firewall filter applied on the DREN IPv6 external peering interfaces as of July 2005.

```
akohli@sdp1.arlapg> show configuration firewall family inet6 filter in_Internet-v6
/* Slammer - MS-=SQL-worm - ticket 240 */
term Block-MS-SQL-worm {
    from {
        packet-length [ 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392
393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414
415 416 417 418 419 420 421 422 423 424 ];
        next-header [ udp tcp ];
        destination-port 1434;
    }
    then {
        count Block_SQLWorm;
        log;
        discard;
    }
}
/* Slammer - NACHI Worm - ticket 2072  */
term Block-NACHI-worm {
    from {
```

```
        packet-length 92;
        icmp-type [ echo-request echo-reply ];
    }
    then {
        count NACHI;
        discard;
    }
}
/* MODULE 2 - TEMP BLOCK OF SOURCE PORT 4000 - TKT 2655 */
term Block-W32-WITTY-WORM {
    from {
        next-header udp;
        source-port 4000;
    }
    then {
        count w32-witty-worm;
        discard;
    }
}
/* Block Bad Ports - HPC Cert - ticket 2195,2675  */
term Block-BadPort-HPCCert {
    from {
        next-header [ tcp udp ];
        port [ 17300 3127 3128 10080 ];
    }
    then {
        count Block-BadPort-HPCCert;
        log;
        discard;
    }
}
/* Module 5A: Sources and Destinations - reject prefixes */
inactive: term reject-martians {
    from {
        prefix-list {
            martian-prefixes-v6;
        }
    }
    then {
        count BLOCK_Martians-v6;
        log;
        discard;
    }
}
/* Module 6: Explicit Port Denials - ICMP */
term Block-ICMP {
    from {
        next-header icmp;
        icmp-type [ redirect router-advertisement router-solicit parameter-problem ];
    }
    then {
        count Block_ICMP;
        log;
        discard;
    }
}
```

```
/* Module 6: Explicit next-header Denials - Cisco Vulnerability */
term Block-Cisco-protocols {
    from {
        next-header [ 53 55 77 ];
    }
    then {
        count Block_Cisco;
        log;
        discard;
    }
}
/* Module 6A: Explicit Port Denials - TCP Source Ports */
term Block-TCP-src-ports {
    from {
        next-header tcp;
        source-port [ 0 69 135 445 555 1433 1434 4444 6101 6661-6669 7100 60001 65000 65535
5434 ];
    }
    then {
        count Block_TCPSrc;
        log;
        discard;
    }
}
/* Module 6B: Explicit Port Denials - UDP Source Ports */
term Block-UDP-src-ports {
    from {
        next-header udp;
        source-port [ 0 69 135 445 555 990-999 1433 1434 4444 6101 6661-6669 7100 8998 32771
60001 65000 65535 5434 ];
    }
    then {
        count Block_UDPSrc;
        log;
        discard;
    }
}
/* Module 6C: Explicit Port Denials - TCP Destination Ports */
term Block-TCP-dest-ports {
    from {
        next-header tcp;
        destination-port [ 0 3 5-7 10-12 14-16 18-19 23 26-27 30-36 38-39 42 44-46 48 52 54 56 58-
65 69-79 84-87 92-93 96-99 109 111 124-135 137-176 178 180-188 190-193 195-199 200-209
212-221 223-241 243-250 260-263 266-310 312-387 391-399 400-406 408-416 420-421 423-424
426-432 434-442 445 447-449 451-463 466-499 502-508 511-519 521 523-532 534-539 541-542
545-547 549 551-553 555-562 564-586 588-590 592-599 600-603 605-635 638-699 701-720
731-748 751-776 778-800 802-819 821-822 824-872 874-899 901 902 904-909 910
        911-935 937-944 946-949 951-985 987-988 991 994 997-998 1000-1007 1009-1023 1034
1042 1243 1433 1434 1993 1999 2049 2222 3127 3128 3389 3917 4444 5554 6000-6063 6129
6500 6661-6669 6711 6712 6776 7000 7100 7350 9898 10008 10080 12345 12346 16660 20034
27374 27665 31337 32777 33270 39168 47017 65000 65535 5434 101 105-108 112 114 116
117 120-122 252-255 722-729 996 999 10000 ];
    }
    then {
        count Block_TCPDest;
        log;
```

```
        discard;
    }
}
/* Module 6D: Explicit Port Denials - UDP Destination Ports */
term Block-UDP-dest-ports {
    from {
        next-header udp;
        destination-port [ 0 3 5-7 10-12 14-16 18-19 26-27 30-36 38-39 42 44-46 48 52 54 56 58-65
69-79 84-87 92-93 96-99 109 111 124-135 137-176 178 180-188 190-193 195-199 200-209 212-
221 223-241 243-250 260-263 266-310 312-387 391-399 400-406 408-416 420-421 423-424
426-432 434-442 445 447-449 451-463 466-499 502-508 511-519 521 523-532 534-539 541-542
545-547 549 551-553 555-562 564-586 588-590 592-599 600-603 605-635 638-699 701-720
731-748 751-776 778-800 802-819 821-822 824-872 874-899 901 902 904-909 910 911-935
        937-944 946-949 951-985 987-988 990-999 1000-1007 1009-1023 1042 1243 1433 1434
1524 1993 1999 2049 3127 3128 3389 3917 4444 4448 5554 6000-6063 6129 6661-6669 6711
6712 6776 7100 7350 8998 10080 16660 27374 27444 31335 31337 32771 32777 41523 41524
65535 5434 101 105-108 112 114 116 117 120-122 252-255 722-729 996 999 ];
    }
    then {
        count Block_UDPDest;
        log;
        discard;
    }
}
term Block-tcp {
    from {
        next-header tcp;
        port [ 91 407 417 604 820 823 ];
    }
    then {
        count Block-tcp-in;
        log;
        discard;
    }
}
term Block-udp {
    from {
        next-header udp;
        port [ 1 2 4 8 9 13 17 20-22 24-25 29 37 40 41 43 47 57 66 82 89 90 91 100 103 104 113
115 118 119 143 177 189 194 222 242 251 264 265 311 388 417 418 419 422 425 433 444 446
447 448 450 451 465 509 510 520 522 533 543 544 548 550 563 591 604 700 721 730 749 801
823 873 900 903 936 945 950 999 1000 ];
    }
    then {
        count Block-udp-in;
        log;
        discard;
    }
}
/* Module 9: Accept All Else - No Sampling */
term Accept-All-Else-NoSample-next-headers {
    from {
        next-header [ 50 51 ];
    }
    then accept;
}
```

```
/* Module 9: Accept All Else - No Sampling */
term Accept-All-Else-NoSample-Ports {
    from {
        destination-port [ 22 25 80 110 443 ];
    }
    then accept;
}
/* Module 9: Accept All Else */
term Accept-All-Else {
    then {
        port-mirror; ## Warning: 'port-mirror' is deprecated
        accept;
    }
}
```

4. ***IPv6 filter applied on the interface (Juniper)***

Same filter attributes (ports and protocols) are configured in the out_Internet_v6 filter as in the in_Internet-v6 firewall filter, This is configured for the external network interface. For example –

```
bjones@hostname> show configuration interfaces fe-2/0/3
 description "IPv6 connection to External network via FE";
 unit 0 {
     description " IPv6 connection to External network via FE ";
     }
     family inet6 {
         filter {
             input in_Internet-v6;
             output out_Internet-v6;
         }
         address xxxx:xxxx:x:xxxx::x/xx;
     }
 }
```