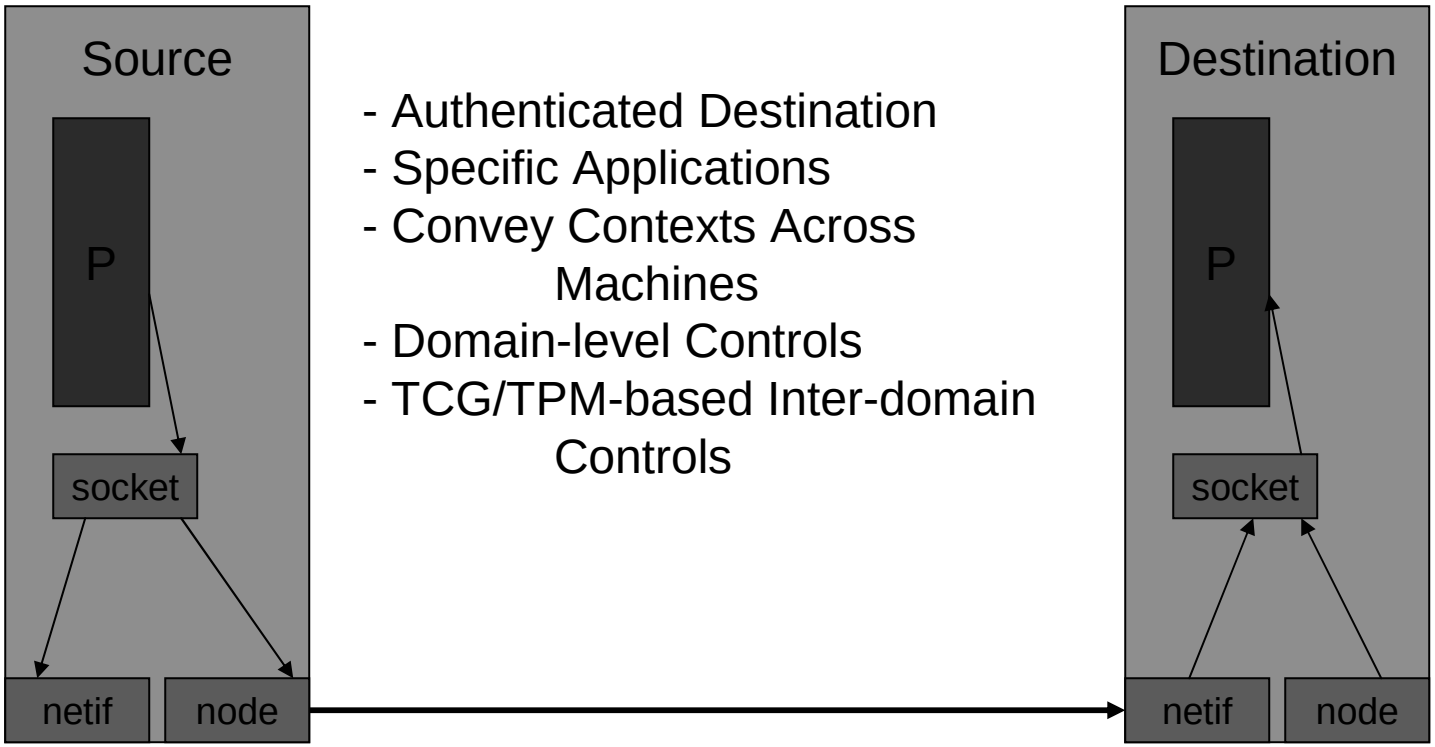
IBM TJ Watson Research Center

Leveraging IPSec for Network Access Control for SELinux

Trent Jaeger – IBM Research, Watson
Serge Hallyn, Joy Latten, George Wilson – IBM Linux Technology Center, Austin

Problem: Network Control

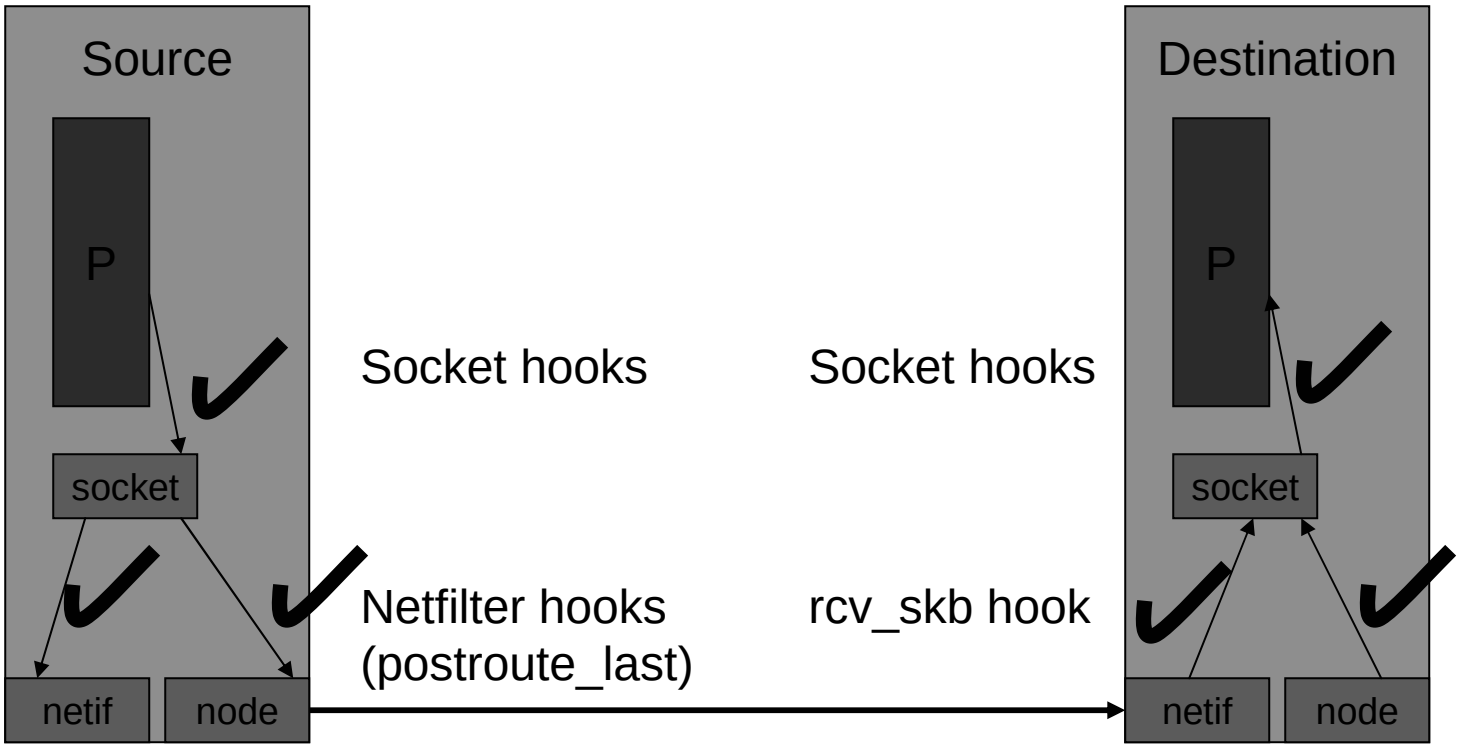


- Authenticated Destination
- Specific Applications
- Convey Contexts Across Machines
- Domain-level Controls
- TCG/TPM-based Inter-domain Controls

Problem: Packet-level Access Control

- Need to ensure that each packet is delivered to authorized subjects
  - Input:
    Each packet could be from arbitrary source
    Need to demultiplex packet to authorized sockets
  - Output:
    Destination and source pair are understood
    Need to convey contexts to destination
- Restrictions on per-packet authorization
  - No significant impact on unlabelled, critical path performance
    IP Security Options too expensive
  - No skb header modifications (performance)
  - No code flow impact
- Granularity of control is not really per packet

Linux Security Modules + SELinux Controls

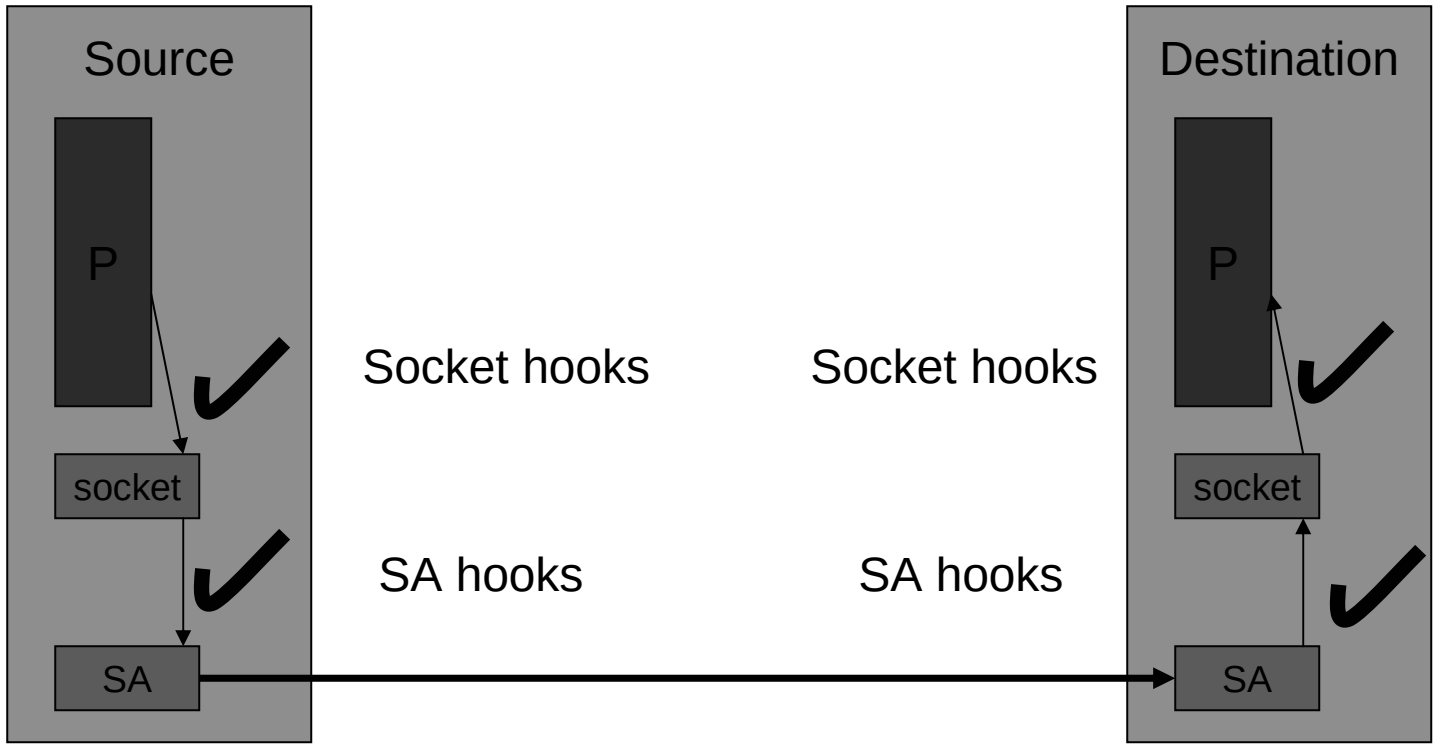


IP Security Options hooks were rejected

IPSec

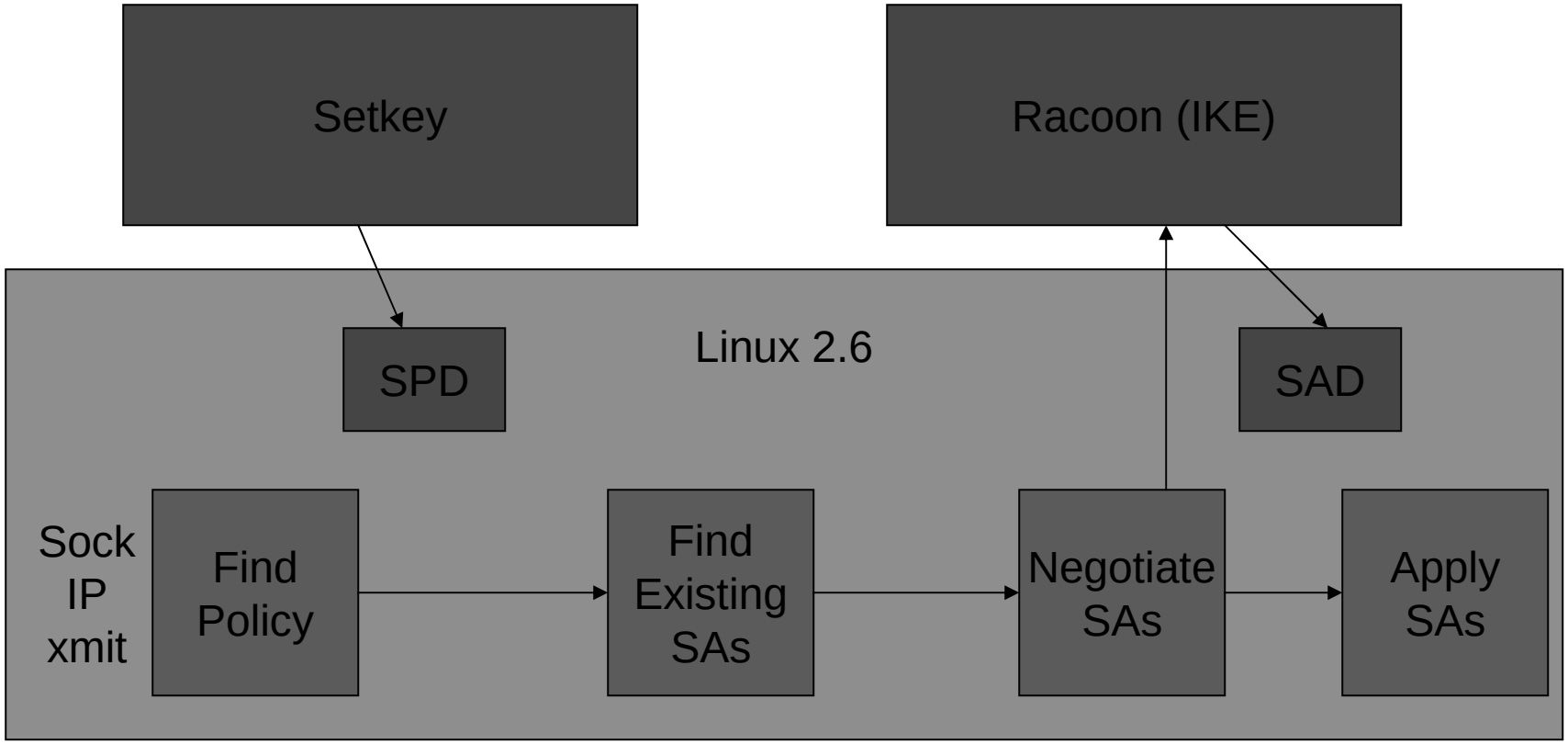
- Privacy and authentication services at the IP layer
  - IPv4 and IPv6
  - Protocols: ESP and AH
  - Key management: PFKEYv2, manual (setkey); IKE, automatic (racoon)
  - Paths: host-host, gateway-gateway, host-gateway
  - Transport or tunnel: single or multiple layers of security protocols
- Security Policy
  - Defines security protocols, mode for source-destination (port)
  - Input to negotiation
- Security Associations
  - Simplex representation of IPSec connection
  - Per protocol (AH or ESP)
  - One mode (transport or tunnel)

Flask and IPSec



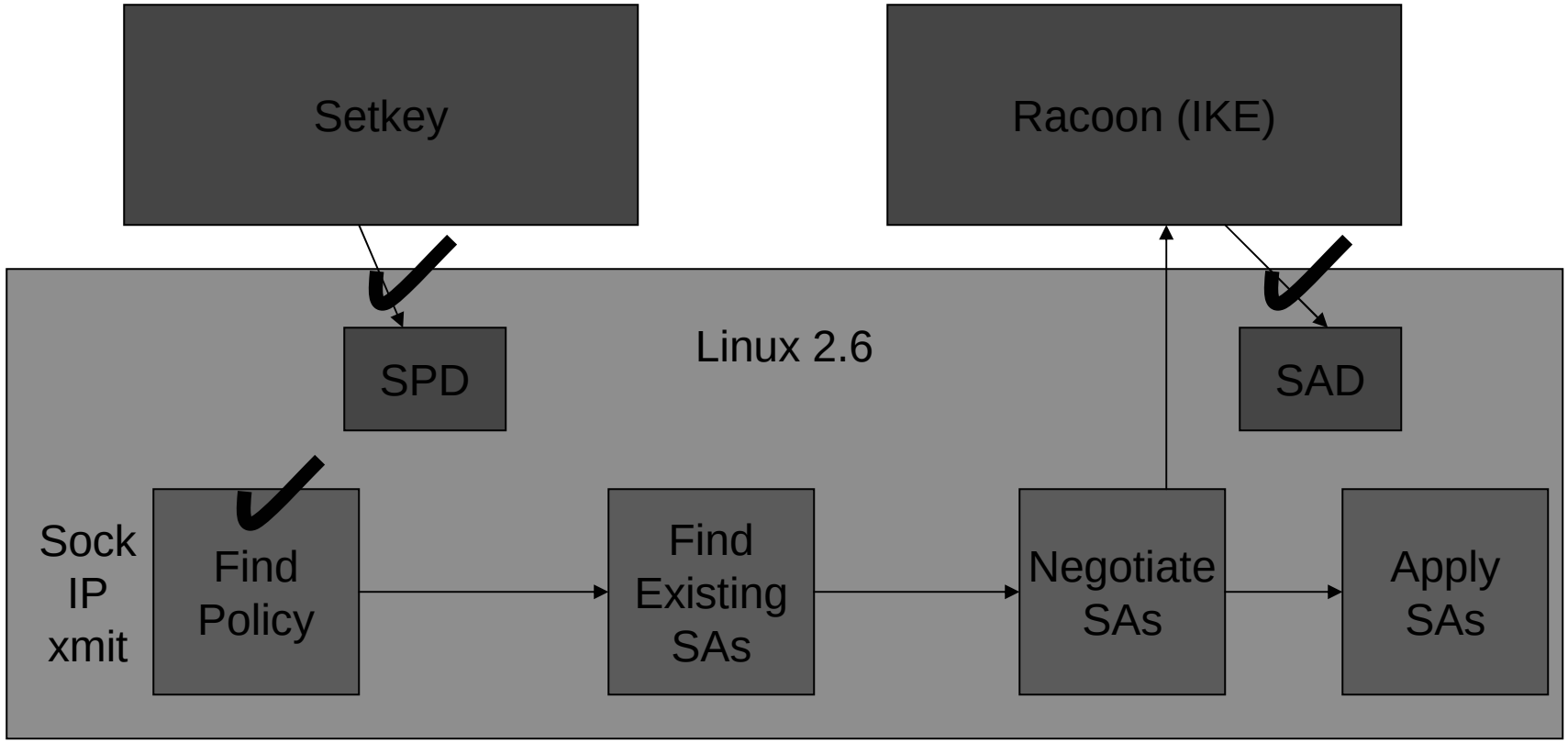
SA is a labeled object; socket labels determine whether can read or write

IPSec protocol – IPSec Tools/Linux XFRM (output)



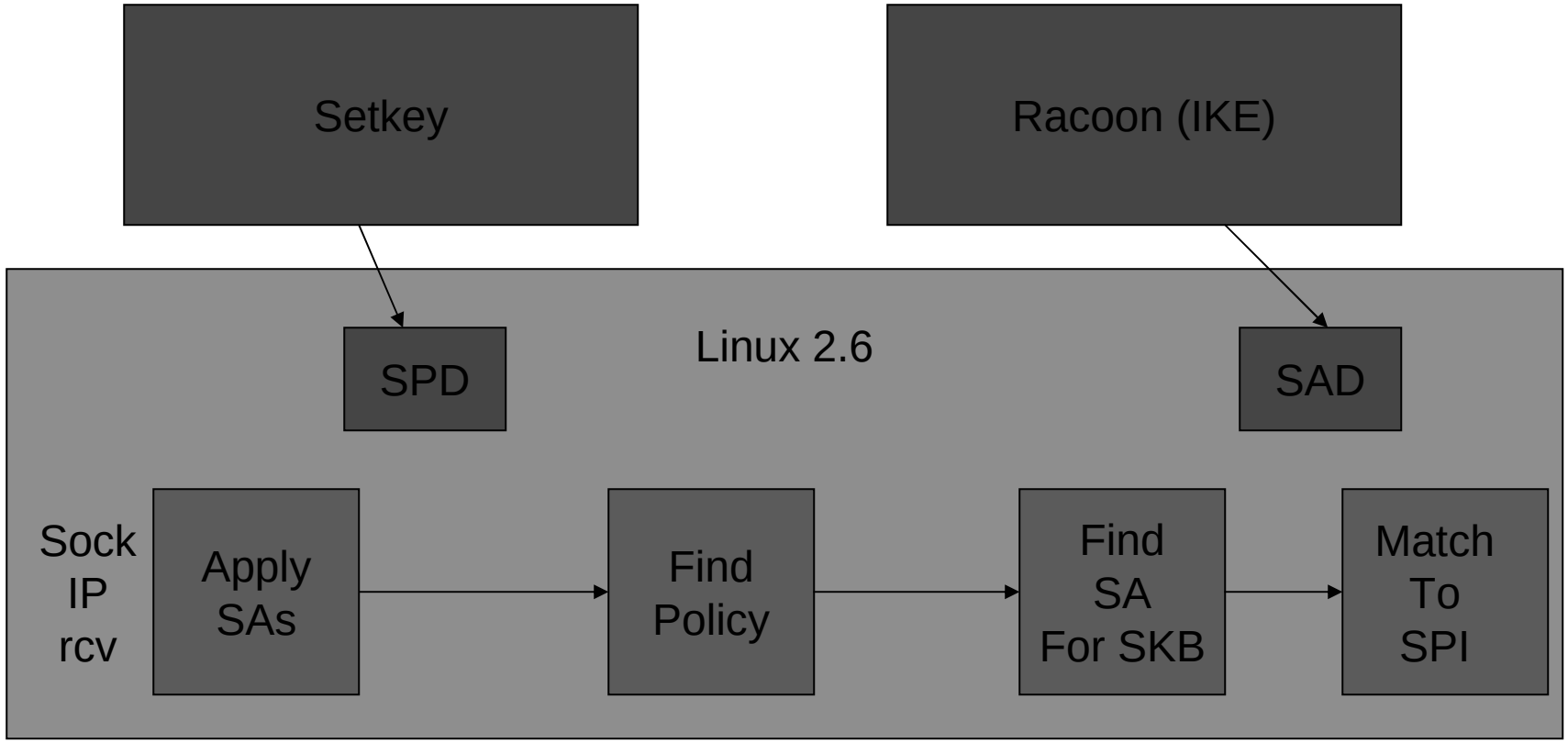
Output: (sk_)policy_lookup → find_bundle → tmpl_resolve → ip_queue_xmit/dst_output

New LSM Hooks (output)



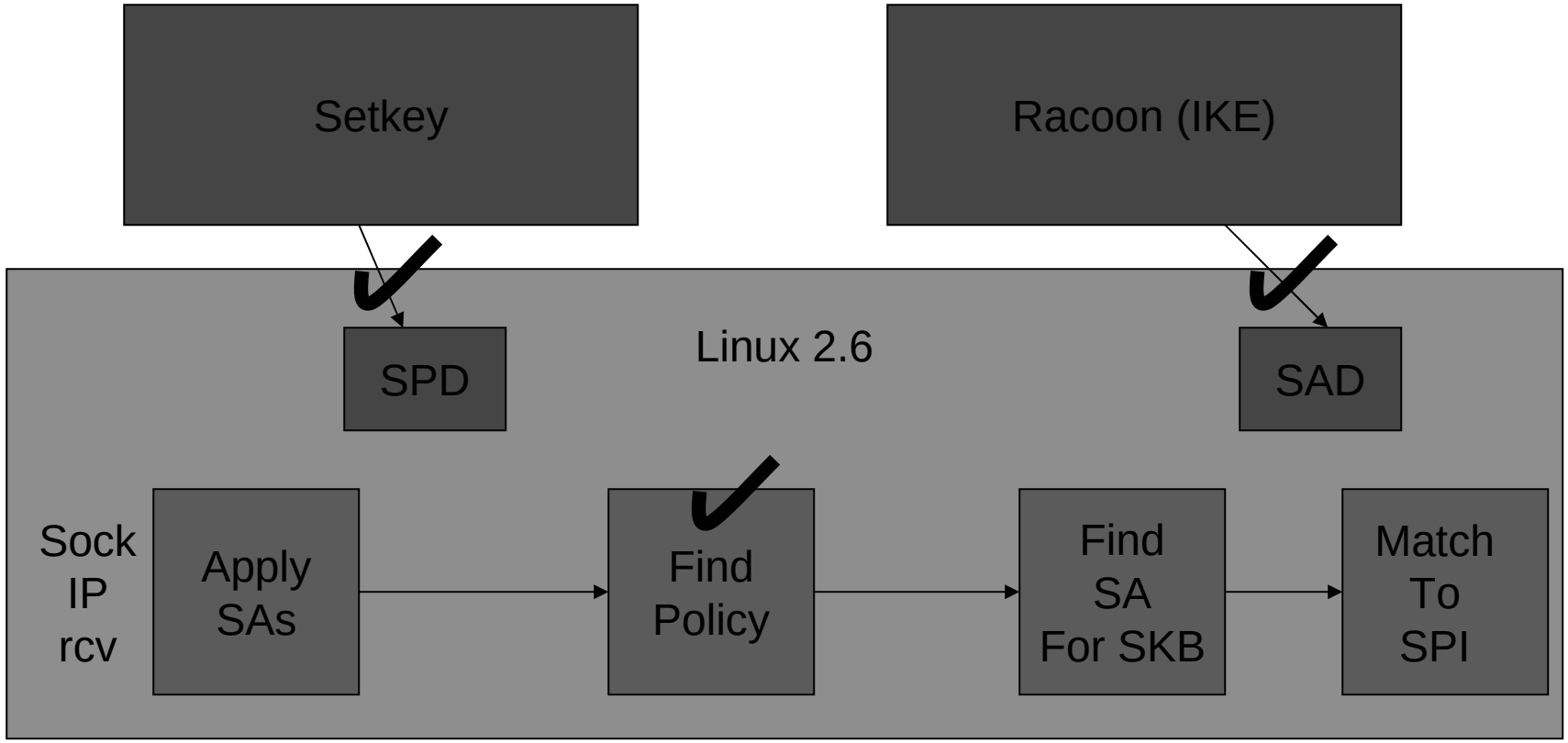
Output: (sk_)policy_lookup → find_bundle → tmpl_resolve → ip_queue_xmit/dst_output

IPSec Protocol: IPSec-Tools/Linux XFRM (input)



Input: ip_rcv_finish/dst_input → (sk_)policy_lookup → policy_ok → state_ok

New LSM Hooks (input)



Input: ip_rcv_finish/dst_input → (sk_)policy_lookup → policy_ok → state_ok

New LSM Hooks and SELinux Implementations

- xfrm_policy_alloc
  - Allocate security data structure in new xfrm_policy
  - Done when policy is added to the SPD (under xfrm_selector)
  - Consists of xfrm_sec_ctx
    - Domain of interpretation
    - Algorithm
    - Context length (string length)
    - Security ID
    - Context String
- xfrm_policy_lookup
  - Authorize socket's use of policy with SA security context
  - Only retrieve/build SA's with the security context of the policy
- xfrm_state_alloc
  - Allocate security data structure in new xfrm_state
  - Done when SA is added to SAD (under xfrm_selector – via xfrm_sec_ctx as SPD)
- New SELinux class
  - Association
  - Operations: sendto/recvfrom
  - allow kernel_t ping_t:association { sendto recvfrom };

Setkey Policy Changes

- **Setkey SPD entries**

```
spdadd 9.2.9.15 9.2.9.17 any -ctx 1 1 "system_u:object_r:zzyzx_t"
        -P in ipsec esp/transport//require ;
spdadd 9.2.9.17 9.2.9.15 any -ctx 1 1 "system_u:object_r:zzyzx_t"
        -P out ipsec esp/transport//require ;
```

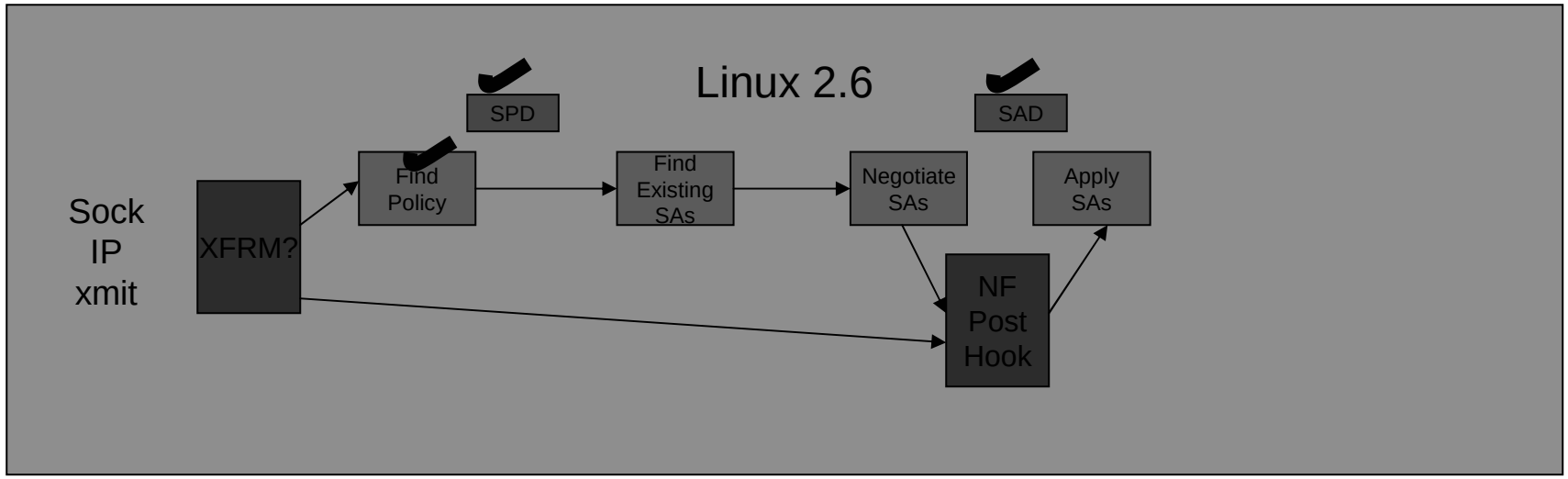
- **Setkey SAD entries (optional as racoon can negotiate)**

```
add 9.2.9.15 9.2.9.17 esp 0x123456
        -ctx 1 1 "system_u:object_r:zzyzx_t"
        -E des-cbc 0x0000000000000000;
add 9.2.9.17 9.2.9.15 esp 0x123457
        -ctx 1 1 "system_u:object_r:zzyzx_t"
        -E des-cbc 0x0000000000000000;
```

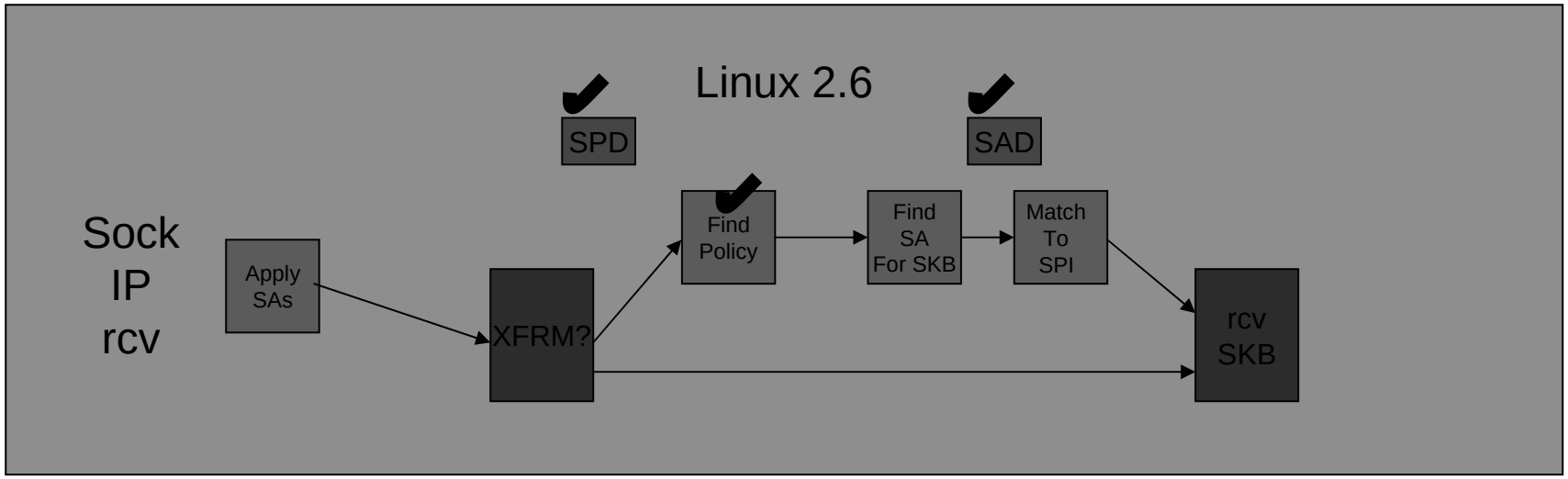
Negotiation model

- Initiator is authorized to only one SA per source-destination-port
  - Granularity of selectors
  - Socket options might distinguish further by per socket policy
    Not currently supported
- Initiator's racoon receives request with policy
  - Authorized to send unlabelled packets only
- Negotiation is a simple context match
  - Types should be same on both sides to indicate same semantics
  - Polyinstantiation
- Each side builds an SA with context
  - Control over sockets that can sendto/recvfrom SA context
- Same context in each direction for racoon
  - Typically, local policies and racoon.conf's are used to generate proposals
    No basis on responder's side to choose counterproposal
  - Verified for encryption and authentication algorithms
  - Does not apply to setkey (manual SA creation)

Overall LSM Network Control (Output)



Overall LSM Network Control (Input)



Overall control (rcv_skb and postroute_last)

```
/* if authorized xfrm, then already authorized
   against xfrm label */
If (authorizable_xfrm_in(skb)) {
        goto accept;
}

/* else, this packet is unlabelled and needs authorization */
sock_sid = get_sock_sid(sk);
rc = avc_has_perm(sock_sid, UNLABELLED, ASSOC,
        op, NULL);
if (rc)
        goto drop;

accept:
drop:
```

Issues

- Policy specification
  - sk_policy vs. manual policy
  - Set by racoon for ISAKMP messages – can use unlabelled
  - Will get rejected unless unlabelled access is allowed
- No sock in some cases
  - E.g., ping and packet forwarding
  - Kernel is the subject in these cases
- Breadth of IPSec use tested
  - Transport for TCP, UDP, ICMP
  - Tunnel
- Patch acceptance
  - 15 files modified: 5 security; 5 net; 5 includes
  - IPSec-tools patch supports these changes
- Inter-system policy management
  - Single domain policy distribution to setkey (in addition to SELinux policy)
  - Cross-domain limited policy use (e.g., based on TCG measurements)
  - Applications use SSL

Summary

- Aim: Network control based on strong authentication on each packet
- IPSec is the kernel service that supports network control
  - XFRM IPSec implementation in Linux 2.6
- Integrate IPSec with LSM and SELinux
  - Control selection of policy for a socket
    Propagated throughout SA retrieval/construction
  - IPSec-Tools modified to support the policy and SA contexts
    Manual (setkey) and dynamic (racoon)
- Intrusiveness to critical path is minimal
  - 1 new LSM hook on IPSec per packet processing – 2 offline
  - 1 more SELinux authorizations for SA in rcv_skb and Netfilter