A Secure PLAN
(extended version)

Michael Hicks, Cornell University
Angelos D. Keromytis, Columbia University
Jonathan M. Smith, U. Penn

DARPA Active Networks Conference and Exposition (DANCE), San Francisco CA, May 29-30, 2002
Nugget to take home:

- A careful separation between language protections for active packets and cryptography-based authorizations for active extensions can lead to a system with flexibility, performance and security

- http://www.cis.upenn.edu/~switchware/PLAN
PLANet: 2-level Architecture

PLAN packet

Install

Routing

...
Trust Management

Joe

OK

Service install
Trust Management

Ed

Service install

not allowed
Form of Service Policies: Access

PLAN packet

Routing

Sending packets

default user
Form of Service Policies: 

**Access**

- Joe
- Install
- Routing
- Sending packets

privileged user
Form of Service Policies: Usage

PLAN packet

Sending packets

default user
Form of Service Policies: Usage

Joe

Sending packets

privileged user
Security Procedure

- PLAN packet
- Routing
- Sending packets

arrival as default user
Security Procedure

Joe

Routing

Sending packets

authentication
Security Procedure

namespace adjustment

Joe

Install
Routing
Sending packets
Security Procedure

... if Joe then more bandwidth
...

per-usage adjustment
Security Mechanisms

- Authentication via HMAC-SHA1
  - signed Diffie-Hellman, as with IPsec
- Authorization Policies - Query Certificate Manager (QCM)
  - language based on sets
  - set descriptions may be distributed
  - Use of certificates for push-based policy
Chunks - units of authentication

- Unit of evaluation in PLAN
  → like a suspended function call

- First-class
  → can be manipulated as data within PLAN programs

- Useful programming construct
  → encapsulation via eval
Chunks - in PLAN packets

Fixed fields

Chunk

evalDest ... code args fun
Ping packet

fun ping(source, dest) =
  ...
fun ack() =
  print("Success")
Core Service

authEval: ‘a chunk * sign -> ‘a

- takes a chunk and an HMAC digital signature and authenticates the chunk
- if successful, performs namespace adjustment and evaluates the chunk
Application: An Active Firewall

- Rather than filter external packets, restrict their privilege
- Accomplished by encapsulating incoming packets with service-restricting chunk

```hs
fun wrap(c,sign) =
  (zeroRB(); authEval(c,sign))
```
Experimental Setup

S

F

D

Trusted network

Untrusted network
Outgoing Ping

\[ S \rightarrow F \rightarrow D \]

ping
Returning Acknowledgement

Firewall signs as and encapsulates packet chunk
Firewall-wrapped Ping packet

```
fun wrap(c, sign) =
  (zeroRB();
   authEval(c, sign))
```
Firewall Performance

Elapsed Time (ms)

no payload  max payload

- Ping
- +Firewall
Firewall Overhead Breakdown
Related Work

- **AN Security**
  - Security Architecture for AN
  - SANE
  - SQoSH/RCANE

- **Language-based protection schemes**
  - SPIN (Modula-3), MMM (Caml), J-kernel (Java), PCC and TAL (x86, Alpha assembly)

- **Trust Management**
  - Keynote, PolicyMaker
Conclusions

- Security in AN: PLANet
  - while preserving performance, flexibility and usability

- Achieved with 2-level architecture
  - language safety in the packets
  - trust management for services

- Useful
  - active firewall (active encapsulation)
Acknowledgments

- DARPA (of course!)
- Scott Nettles, Jonathan Moore, Scott Alexander, Bill Arbaugh, Trevor Jim, other SwitchWare team members
- An earlier (and highly abbreviated) version of this paper was presented at IWAN99 in Berlin, July 1999
Questions and Discussion
SwitchWare System Architecture

PLAN
ALIEN/Caml/OS
SwitchWare System Architecture

ALIEN Library

PLAN Packet
Caml Active Code
Caml Active Code

ALIEN/Caml/OS
AEGIS
ALIEN Active Loader

D. Scott Alexander’s Ph.D. thesis

- Libraries
- Core Switchlet
- Loader
- Runtime (Caml)
- OS (Linux)

Protection Boundary
Mutability Boundary