Multidomain Virtual Security Negotiation over the Session Initiation Protocol (SIP)

1st International Workshop on Critical Information Infrastructures Security

August 31st - September 1st 2006. Samos, Greece
Contents

- Introduction
- The SIP Protocol
- Security Negotiation over SIP
- Integration with Policy Based Network Management
- Testbed
- Conclusions
- Future Work
Introduction
The DESEREC Project

- IST Project: IST-2004-026600
- Target: “to increase the dependability of critical open and interconnected information systems by a multidisciplinary, coordinated effort”.
- Website (under construction): www.deserec.org
Introduction
Dynamic Security Overlays

- Different organizations usually have different:
  - Security policies
  - Security technologies
  - ...

- Security overlays allow defining an abstract layer on the top of the actual deployment
Introduction
Dynamic Security Overlays

- Such overlays are usually static, requiring static pre-configuration
- Dynamic security overlays would require an automatic negotiation process between domains
  - Virtual, abstract specification of the security requirements for the negotiation
  - Mapping from virtual requirements to actual security technologies
The SIP Protocol

- SIP (Session Initiation Protocol) is a signalling protocol for establishing data sessions
- SIP allows peers to negotiate session parameters
- Other advantages:
  - It is a standard protocol (see RFC 3261)
  - It is extensible through definition of custom headers
  - It allows routing call establishment dialogs through several different domains, by relying on intermediate entities (SIP proxies)
  - Can be easily secured (TLS, S/MIME)

→ It is well suited for performing a multidomain negotiation
The SIP Protocol

- There are two kinds of SIP elements:
  - User agents, which demand the establishment of a session between them
  - Proxies, which route SIP calls across domains
- Location of SIP elements:
  - One user agent at each end-user: for stating the initial requirements, and accepting or rejecting the other peer’s offering
  - One proxy at each domain: for routing the call, and perform domain-to-domain negotiation
The SIP Protocol

Typical 2-domain SIP scenario:
Security Negotiation over SIP

- Dynamic Security Overlays require a two-level negotiation:
  - The involved domains need to agree a common set of security requirements, expressed in any way which is consistently understood by all of them
    - \{confidentiality=1, integrity=5, accessControl=3\}
  - Security technologies (e.g. IPsec, SSL, etc.) must be agreed for use between each pair of consecutive domains
- This forms the so called virtual security negotiation process
Security Negotiation over SIP
Security Negotiation over SIP

- Custom headers can be defined for SIP messages
- SIP user agents and proxies use these headers and process them to implement a custom behaviour during the negotiation
- This is the basis for extending SIP in order to customize the negotiation process
  - Can be used to implement the virtual negotiation
Security Negotiation over SIP

Three main goals:

1) **Ensure that requirements are acceptable.** Each node (except the one initiating the call) must check that it can support the requested virtual security. If not, it must cancel the call. It may update these requirements if necessary, but without violating the current *max* and *min* limits.
Three main goals (continued):

2) **Consecutive domains must agree on a technology.** This will be done via an *offer* mechanism: one node will state its available technologies, and the next one will choose one that it supports too (and notify about which). If this agreement is not possible, the call must be cancelled.

3) **Some domains might be not compliant.** These domains must be detected, and bypassed with a tunnel.
Security Negotiation over SIP

Custom headers added:

- **X-MinSecurity, X-MaxSecurity**: security requirements which an end user desires
- **X-Via**: list of compliant domains traversed
- **X-Tunnel**: tunnel endpoints for bypassing non-compliant domains
- **X-TechList**: list of security technologies (IPSec, SSL, …) supported for implementing the requirements
- **X-SelectedTech**: chosen security technology for using between consecutive domains
Integration with Policy Based Network Management

- Our SIP entities may need domain-specific information at several points:
  - Which maximum and minimum security level a user is allowed to get?
  - Which security technologies are available?
  - …
- All of this information may be retrieved from a Policy Based Network Management system (PBNM)
Integration with Policy Based Network Management

- A PBNM system usually comprises:
  - A policy database
  - One or more policy servers (PDP’s)
  - One or more policy clients (PEP’s)

- SIP entities can be policy clients as well (embedding a PEP), so that their behaviour is bound by the domain policies defined by the administrator
Testbed

- A sample testbed was implemented for demonstrating the developments:
Testbed

Alice’s virtual security requirements are in the X-MinSecurity and X-MaxSecurity headers.
Testbed
Testbed

- Alice
- P1
- P2
- Bob

**DOMAIN 1**

**DOMAIN 2**

1) Check requirements;
2) Update requirements;
3) Build an *X-TechList* header with available technologies.
Testbed

1) Check requirements; 2) Choose one technology from X-TechList header (“techno1”); 3) Replace X-TechList header with own.

1) Updated requirements in X-MinSecurity and X-MaxSecurity headers; 2) Bob’s available technologies in X-TechList header.
Testbed

1) Check requirements;
2) Choose one technology from $X$-TechList header (“techno2”);
3) Replace $X$-TechList header with own.
Testbed

1) Choose one technology from X-TechList header ("techno3");
3) Build an X-SelectedTech header with the value "techno3".
Testbed

1) Carries the $X$-SelectedTech header with the value “techno3”; 2) After sending the ACK, Alice instantiates the “techno3” technology.

1) Read “techno3” from the $X$-SelectedTech header; 2) Update that header, replacing “techno3” with “techno2”.

University of Murcia

CRITIS 2006
Samos, Greece
Testbed

1) Carries the X-SelectedTech header with the value “techno2”; 2) After sending the ACK, P1 instantiates the “techno3” and “techno2” technologies.

1) Read “techno2” from the X-SelectedTech header; 2) Update that header, replacing “techno2” with “techno1”.

DOMAÍN 1

DOMAÍN 2
1) Carries the X-SelectedTech header with the value "techno1"; 2) After sending the ACK, P2 instantiates the "techno2" and "techno1" technologies.

1) Read "techno1" from the X-SelectedTech header; 2) Instantiate the "techno1" technology.
Testbed

Domain 1

techno3

P1

Domain 2

techno2

P2

techno1

Alice

Bob
Conclusions

- We managed to implement dynamic security overlays, using:
  - An extension of SIP. Additional headers created, but existing ones were not modified (backwards compatible with legacy SIP proxies)
  - Negotiation of abstract, virtual security requirements, which are dynamically mapped to actual technologies at each step
  - Whole process can be policy-driven
Future Work

- Addition of bootstrapping mechanisms
  - Mobile users
- Integration with real, high level applications
  - Multimedia, peer to peer, ...
- Enhance the semantics of the custom headers
  - Automatic detection of end-to-end support of security technologies, reducing the overhead on intermediate nodes
Multidomain Virtual Security Negotiation over the Session Initiation Protocol (SIP)

Thank you for your attention!