Research Topics – Security and Device Authentication

Industrial systems for all verticals like energy, transportation, automation systems, health require secure device authentication and machine-2-machine communication as a core security feature. Siemens is a leading provider for technology and solutions in these areas.

Situation

- Machine-2-Machine connectivity down to field devices is a major driver for Future Internet. Potentially more than 60 Billion devices will be connected. All industries are affected including manufacturing, process, building, energy automation, transportation, health.
- Device authentication is the prerequisite to ensure an appropriate protection of communication between different devices. This is the basis to realize secure device-oriented services like secure control, monitoring, remote service, metering, licensing or anti counterfeiting.
- Device credentials (keys, certificates) have to be generated and managed efficiently
- The non-human security environment requires new device-oriented security and identity infrastructures.
- The huge number of devices and the specific application environment require a zero-configuration effort.
- Process comprises the generation (short and long term), distribution and implementation of initial security parameter of the target devices

Topics

- Efficient cryptographic mechanisms for device authentication and secure device communication
- device security platform module
- device-oriented security and identity infrastructure (processes, scalability, limits of authority, privacy)
- Plug-and-play security to avoid administrative burden
- secure device-oriented services

prof. Beniamino Di Martino
Project Coordinator mOSAIC Project
Dipartimento di Ingegneria dell'Informazione
Seconda Università degli Studi di Napoli

Portability and Interoperability in Cloud Computing

Beniamino.dimartino@unina.it

www.mosaic-cloud.eu

SEVENTH FRAMEWORK PROGRAMME

Cloud Computing

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.

NIST Definition

Cloud Computing Idea

- Delegate *Everything* to the Network
- Self-Service: User access resource without an human intermediate
- Pay-per-use

Cloud Computing Levels

Software as a Service

Applications

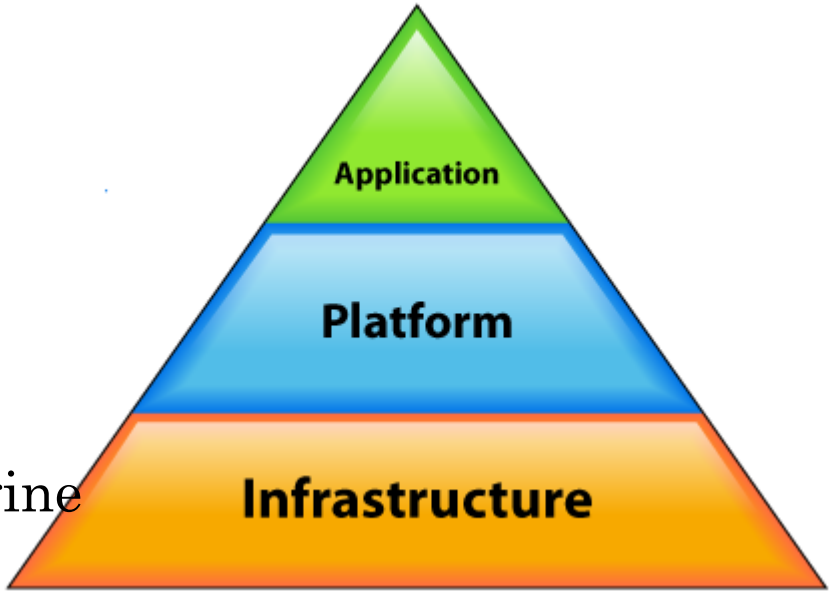
Example: Google Docs

Platform as a Service

Development Environment

Examples: Google App Engine

Microsoft Azure



Infrastructure as a Service

CPU, Storage, ...

Example: Amazon EC2

The Cloud Computing 5 Challenges

Cloud Computing is an emerging paradigm, in literature 5 main challenges are identified:

- data and application interoperability
- data and application portability
- governance and management,
- metering and monitoring,
- Security.

We focus on portability and interoperability

The Problems in few Words (1/2)

Imagine that you are responsible for your corporation's data center, and you experience wild variability in the computational load in your data center. You only have jobs to run at the end of every month, in the rest of time your infrastructure is almost unused. Still, you have high maintenance costs.

You heard about Cloud computing earlier, you took a two days course, and you are made to believe that it will solve your problems. When you introduce Cloud services to your organization, however, you realize this task is not as easy as it seemed.

The Problems in few Words (2/2)

First, you realize that you should use different Cloud providers each month, since the variable computational requirements fit different Clouds offer.

The second issue is that you cannot find a single Cloud provider that has all the required services.

If you think that the above description fits you, then the mOSAIC platform is for you. Using mOSAIC solution you do not have to decide on a specific Cloud provider at design time, instead any time you use Cloud services, you will access the ones fitting best your needs.

mOSAIC Approach

The mOSAIC project (www.mosaic-cloud.eu) aims to develop an open-source platform that enables applications to negotiate Cloud services as requested by their users.

Using the Cloud ontology, applications will be able to specify their service requirements and communicate them to the platform via the innovative API.

The platform will implement a multi-agent brokering mechanism that will search for services matching the applications' request, and possibly compose the requested service if no direct hit is found.

mOSAIC Approach

Cloud-application developers and maintainers will be able to postpone their decision on the procurement of Cloud services until runtime, while end-user applications will be able to find best-fitting Cloud services to their actual needs and efficiently outsource computations.

The platform will facilitate competition between Cloud providers, who, in return, will be able to reach customers they could not reach before.

Security & Trust

Panel highlight on security and trust within the Internet

Background

- In the past several months a series of highly sophisticated and targeted cyber attacks has revealed a shift in the threat arena and their persistence on networks (APT buzz)

- Attackers are moving beyond schemes to acquire financial data (such as credit cards and identity theft) and are pursuing high-value digital assets such as intellectual property, access to critical operations, and other proprietary data systems

- We are involved with businesses in attempt to review possible strategies and validation of business models that aim to offer mitigation against a portion of this space.

Increased Effectiveness - Social Networking

- At the end of 2010, nearly 85 percent of recorded phishing attempts used social networks as a lure, up from 8.3 percent at the start of the year

- Tiny URLs have also enabled better hiding of phishing domains from users. Phishing detection software generally looks for HTML-based content, hence some attacks are using Flash, JavaScript and MIME type content that autocorrects to HTML in browsers for success

Social Engineering

- Most recent compromises that are reported use a technique called social engineering
- Defined - social engineering is using deception, manipulation, and influence to convince a human who has access to a computer system to do something, such as click on an attachment or a link in an email
- Social-engineering schemes historically use spoofed e-mails purporting to be from legitimate businesses and agencies to lead consumers to counterfeit websites designed to trick recipients into divulging financial data such as usernames and passwords.
- Recent attacks now bundle this method via spear phishing, which leads to a much more targeted attack; in order to access to a key person's workstation, data, etc.

Spear Phishing

- Phishing is a mechanism that employs both social engineering and malicious means to steal identity & financial account credentials
- Defined – Phishing is the act of tricking someone into surrendering private information over the Internet, follows the idea of actual fishing — you throw out bait with the hopes that while some ignore it, others will bite.
    - Traditional attacks mimic financial sites to collect credentials or online shopping sites to collect credit card data
    - Attacks most commonly come in the form of emails or messages that contain viral links.
- Recent trends show an increase of compromises that use techniques to allow an attacker access to a key person

Increased Effectiveness - Social Networking

- Social Networks make it easy…..one can go into Twitter, Facebook, LinkedIn, etc. to search for someone or use current events to lure recipients to react to a communication
- Online communities are powerful, trusted and perfect for cyber crime to leverage such relationships
    - Social networking attacks leverage a trusted link between friends, either to deliver malware or to phish for confidential and financial information.
    - Easier for attackers to spread malicious software through links, photos and applications because those users are typically more trusting




Questions Your Organization Should Ask

Q: How many of our users would fall prey to a spear-phishing attack?

Q: Would attackers be able to hijack admin accounts?

Recent examples raise questions about important web defense strategies such as protecting remote users and office-based workers through 24/7 security…

Thoughts

- The real issue isn't the type of mechanism being used to target victims, It's that users are simply not learning how to avoid being tricked on the Internet

Saint Petersburg State Polytechnical University

Software reliability improvement by static analysis methods

Mikhail Glukhikh
glukhikh@kspt.ftk.spbstu.ru

Mikhail Moiseev
mikhail.moiseev@gmail.com

The Fourth International Conference on Dependability
August 2011

ST. PETERSBURG STATE POLYTECHNICAL UNIVERSITY

Software Reliability Problem

- Software systems and components become the most important part of technical systems
- The main cause of software failures is defects in source codes
- For C/C++ programs in average – **0.25defects / 1KLOC***
- The most wide-spread defect types
  - Incorrect pointer dereference – 42%
  - Memory and resource leak – 23%
  - Unreachable code – 10%
  - Use before initializing – 8%
  - Buffer overrun – 6%

* Coverity Scan: 2010 Open Source Integrity Report

Defect detection by Static Analysis

- For defect detection we use static analysis
  - Detection of all defect types
  - Fully automatic analysis process
  - High sound and precision
  - Modest resource consumption
- Our static analysis uses abstract interpretation based on program states
  - Program model building (CFG)
  - Program state extraction
  - Defect detection

Aegis for C/C++

- Automatic defect detection tool – Aegis for C/C++
  - Detection of all primary defect types
  - Full C99 and C++2003 support
- Aegis features
  - Interprocedural, context-sensitive analysis
  - Condition interpretation for branches
  - Complex object analysis
  - Pointer arithmetic analysis
  - Cycles and recursions are supported
  - Analysis of parallel programs

Aegis Characteristics

- Analyzed projects
  - PostgreSQL, FreeBSD, Wine
  - Linux utilities and SourceForge projects
  - Some commercial projects
- Total size of code analyzed is 3 millions of LOC
- Efficiency characteristics
  - Detected defect density is 1defects / 1KLOC
  - Precision is about 35%
  - Analysis time – 10 minutes for projects up to 50 KLOC

Parallel Program analysis

- Aegis C/C++ supports multithreaded programs based on Pthread
- Aegis for SystemC allows detection of synchronization errors in hardware/software system designs
  - Deadlock and Data race detection
  - Analysis of untimed and timed designs
- Future works are related to Worst-Case Execution Time analysis and improvements of static analysis

Defect classification for C/C++ program

#	Name	Description
RES	Incorrect operation with dynamic memory and resources	Multiple release, workflow violation
LEAK	Memory and resource leak	Loss of last reference on dynamic object or resource
BUF	Incorrect operations with buffers and arrays	Buffer overrun, operations with pointers out of buffer
INI	Lack of initialization	Using uninitialized objects
SYN	Synchronization errors	Data races, deadlocks and others

Defect examples

- Incorrect operations with pointers
  - dereference pointer which is out of object bound
  - operation with pointers that pointed to different objects

```
int main()
{
  int arr[10];
  int *p;
  …
  p = arr;
  for(int i=0; i<=10; i++)
     printf("%d" , *p++);
}
```

```
int main()
{
  int a[10];
  int b[15];
  int *p = a;
  int *q = b+10;
  …
  int diff = (p - q);
}
```

ST. PETERSBURG STATE POLYTECHNICAL UNIVERSITY

Peterburg Polytechnical University

Defect examples

- Use before initialization
  - read of uninitialized variable
  - uninitialized pointer dereference

```
int main() {
  int i;
  …
  if (i > 0){
     …
  }
  …
}
```

```
int main() {
  int *p;
  …
  if (flag){
    int i =*p;
  }
  …
}
```

Defect examples

- Deadlock

```
pthread_mutex_t m;

void f() {
  pthread_mutex_lock(&m);
  …
  pthread_mutex_unlock(&m);
}

int main() {
  …
  pthread_create(&t,NULL,f,NULL);
  pthread_mutex_lock(&m);
  …
  pthread_join(t, NULL);
  pthread_mutex_unlock(&m);
  …
}
```

Defect examples

- Data Race

```
int a, b;

void f() {
  a = 1;
}

int main() {
  …
  pthread_create(&t,NULL,f,NULL);
  …
  a = 2;
  …
  pthread_join(t, NULL);
  b = a;       // b = 1 или 2
  …
}
```

WWW.DIGITEKLABS.RU/AEGIS

Главная | Aegis

Результаты анализа

Длительность анализа: 6 seconds and 882 milliseconds

Путь	Код	Дефекты, контексты
	01: // Several for-cycles with function call	
	02:	
	03: #include <stdio.h>	
	04:	
	05: int arr[2];	
	06:	
	07: int f1(float par) {	
	08: int tmp = par;	
	09: printf("%d \n",tmp);	
1(3)	10: return arr[tmp]=tmp; // BUF-02	Разыменование указателя int * arr, tmp выведенного за границу объекта (BUF-02)

Security Challenges and Safety Control in Future Internet

Panel Discussion

Stefan Rass

System Security Research Group
Alpen-Adria University Klagenfurt

Austria

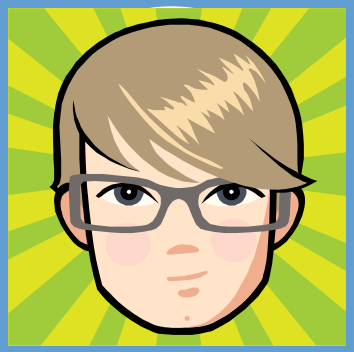
syssec

ALPEN-ADRIA UNIVERSITÄT
KLAGENFURT | WIEN GRAZ

Secret Information Leakage – “Classical Eavesdropping”

Alice

Adversary



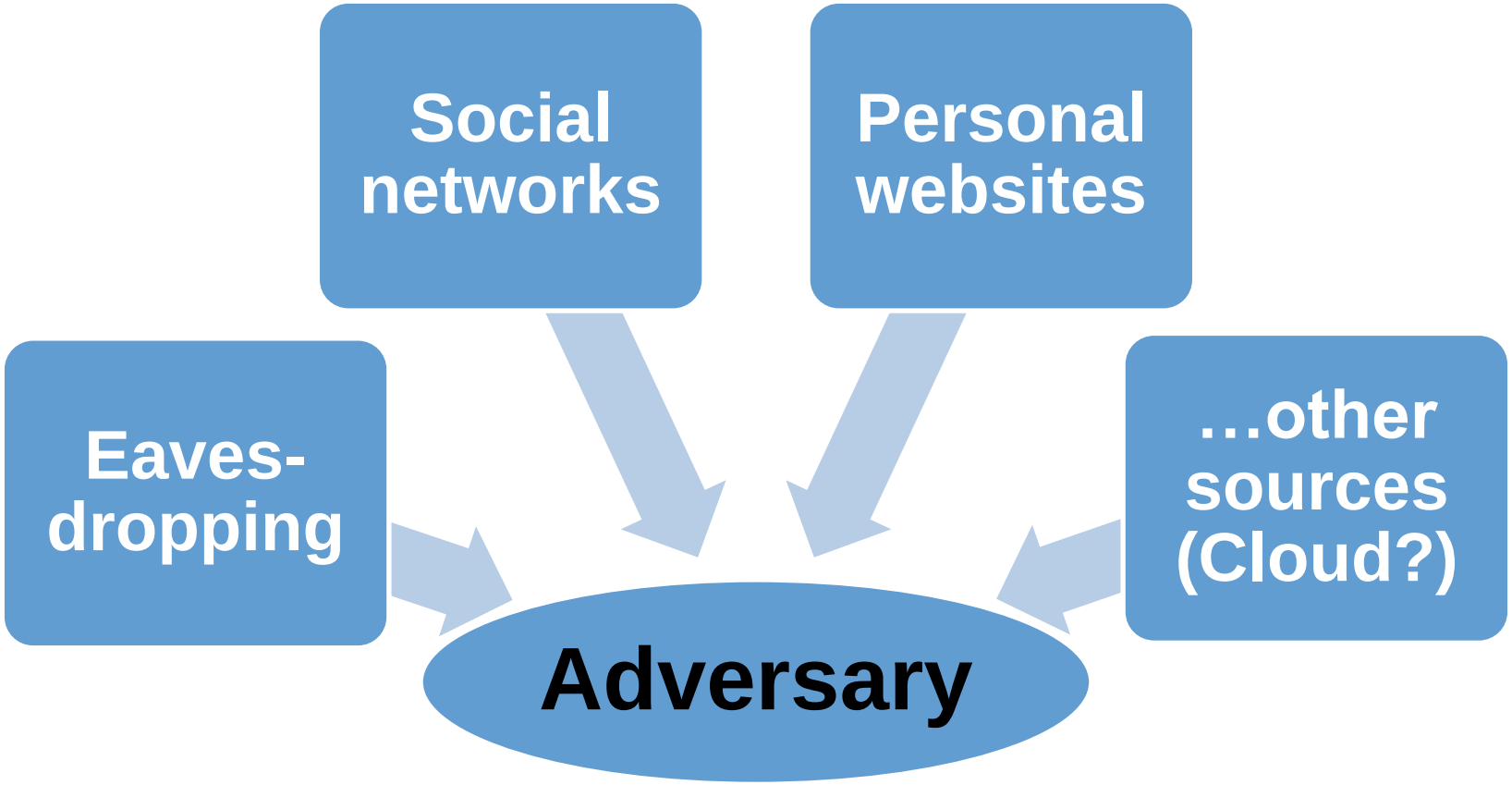
Bob

Cryptographic Channel

syssec

ALPEN-ADRIA UNIVERSITÄT
KLAGENFURT | WIEN GRAZ

Now: Consolidation of diverse sources….



Challenges & Proposals

- User‘s security awareness: …weakest point in the security chain
  → tough to control
- Can infrastructure assist an unaware user?
- Proposed direction for research:
  - Risk-driven infrastructure design
  - Focus on attacker-defender-scenarios
  - Combine Game-Theory and Information-Theoretic Security

Implications & Research

- "Security optimization"
  → Combinatorial Optimization
  → Complexity Theory
- Cryptography offers many qualitative security notions
- What decision-makers look out for: protection of assets with known value
- Compatibility of cryptographic security and decision-theoretic risk? → game-theory.

Links and further information

- The SERIMA project @ http://www.syssec.at/serima/

- Publications related to risk-driven infrastructure design:
  - S. Rass & P. Schartner: **A unified framework for the analysis of availability, reliability and security, with applications to quantum networks**, in IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, 2010, 40, 107-119.
  - S. Rass, A. Wiegele, P. Schartner: **Building a Quantum Network: How to Optimize Security and Expenses**, in Springer Journal of Network and Systems Management, 2010, *18*, 283-299.
- Our research group: www.syssec.at

syssec

ALPEN-ADRIA UNIVERSITÄT
KLAGENFURT | WIEN GRAZ