

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

ORCHIDS, and Bad Weeds

Jean Goubault-Larrecq, Julien Olivain



RV'08 — Budapest, March 30, 2008

Purpose of this talk

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

ORCHIDS is a real-time, multi-event, multi-source intrusion detection system.

- First, we'll talk (a lot) about **computer security**.
- Second, although the initial ideas come from model-checking, ORCHIDS really implements a form of **monitors**, with a few twists.

Outline

- 1 Some Security Issues
- 2 Example: The `ptrace` Attack
- 3 Detecting The Attack, Using ORCHIDS
 - Demo
 - Under the Hood
- 4 Beyond: Additional Features, Further Attacks
- 5 Conclusion
- 6 Other Things That Cannot Fit In The Talk
 - The Architecture of ORCHIDS
 - Way Beyond

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Some Types of Attacks

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

- Viruses, worms, trojans, buffer overflows, etc.
(attacks on systems)
- Denial of service, IP/ARP spoofing, sniffing, etc.
(network attacks)
- Attacks on html forms (perl, pgg), SQL insertion,
IE/Word viruses [worms], etc.
(attacks on applications)

Current Trends

- Systems and networks grow **larger**.

More and more difficult to ensure any level of security.

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Current Trends

- Systems and networks grow **larger**.

More and more difficult to ensure any level of security.

- Stakes are **higher**.

On-line data-bases [banking, health, taxes, ...],
e-commerce, etc.

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

Current Trends

- Systems and networks grow **larger**.

More and more difficult to ensure any level of security.

- Stakes are **higher**.

On-line data-bases [banking, health, taxes, ...],
e-commerce, etc.

- Attacks are more and more **sophisticated**, automated, and distributed.

Ready-to-use packages [ask Google]

Attacks requiring several steps,

... each of them being innocuous in isolation

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

Current Trends

- Systems and networks grow **larger**.

More and more difficult to ensure any level of security.

- Stakes are **higher**.

On-line data-bases [banking, health, taxes, ...],
e-commerce, etc.

- Attacks are more and more **sophisticated**, automated, and distributed.

Ready-to-use packages [ask Google]

Attacks requiring several steps,

... each of them being innocuous in isolation

- **New needs** in intrusion detection

Tracking user configurations

Detecting internal fraud

Smartcards

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

Julien Olivain



Hey, Jean! You're not just
going to blabber along, right?
Show them the `ptrace`
attack for starters.

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo Under the Hood

Beyond

Conclusion

Misc

Architecture Way Beyond

Julien Olivain



Hey, Jean! You're not just going to blabber along, right? Show them the `ptrace` attack for starters.

Er, that's what I meant to do... sure!



Me
(confused)



Outline

- 1 Some Security Issues
- 2 Example: The `ptrace` Attack
- 3 Detecting The Attack, Using ORCHIDS
 - Demo
 - Under the Hood
- 4 Beyond: Additional Features, Further Attacks
- 5 Conclusion
- 6 Other Things That Cannot Fit In The Talk
 - The Architecture of ORCHIDS
 - Way Beyond

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

The ptrace Attack [Purczyński01,03]

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

The ptrace Attack [Purczyński01,03]

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Exploits a problem on rights of spawned processes, with the `ptrace` system call.

Effect: a **local, user-to-root** attack.

The ptrace Attack [Purczyński01,03]

ORCHIDS

ORCHIDS
Laboratoire
de
Sécurité
et
de
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Exploits a problem on rights of spawned processes, with the `ptrace` system call.

Effect: a **local, user-to-root** attack.

- The `ptrace` call: used by **all** debuggers (benign in isolation!). Requires **correlations**.
- Subtle attack, based on a **race condition** in Linux 2.18 (Red Hat) kernels.

The Hacker's View

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

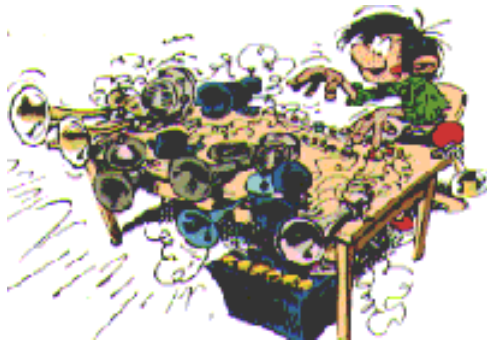
Beyond

Conclusion

Misc

Architecture

Way Beyond



The ptrace Attack [Purczyński01,03]

ORCHIDS

SEC (st) Laboratoire Specification Verification

CVS

INRIA

Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

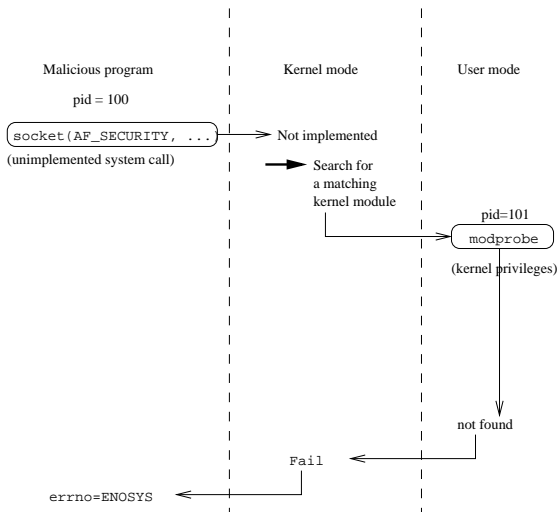
Demo Under the Hood

Beyond

Conclusion

Misc

Architecture Way Beyond



The ptrace Attack [Purczyk01,03]

ORCHIDS

ORCHIDS
Laboratoire
de Sécurité
et de
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

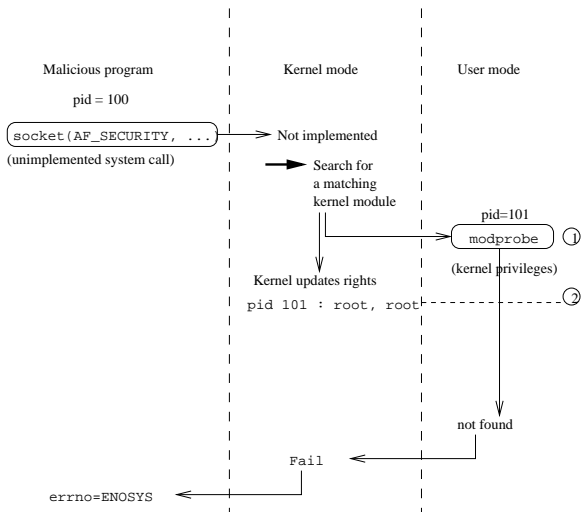
Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond



The ptrace Attack [Purczyński01,03]

ORCHIDS

Laboratoire
Spécification
Vérification

ENS
CNS

INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

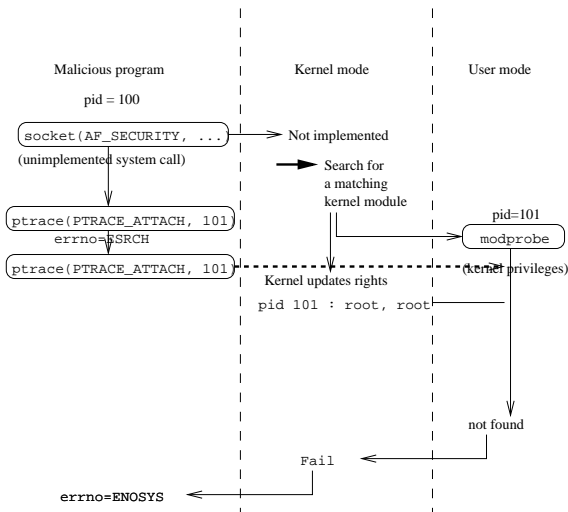
Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond



The ptrace Attack [Purczyński01,03]

ORCHIDS

Laboratoire de Sécurité et de Vérification



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

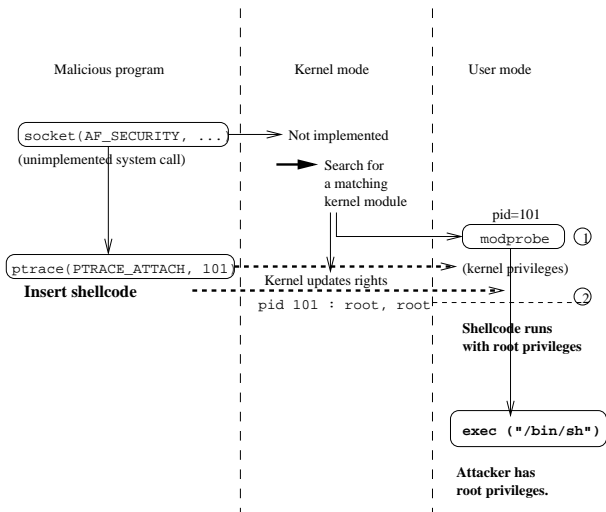
Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond



Outline

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

- 1 Some Security Issues
- 2 Example: The `ptrace` Attack
- 3 Detecting The Attack, Using ORCHIDS**
 - Demo
 - Under the Hood
- 4 Beyond: Additional Features, Further Attacks
- 5 Conclusion
- 6 Other Things That Cannot Fit In The Talk
 - The Architecture of ORCHIDS
 - Way Beyond

The Attack Signature

- We can count on the system **logging** important events.
Here we count on the SNARE kernel module.
We may also interface to the `syslog` facility.

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

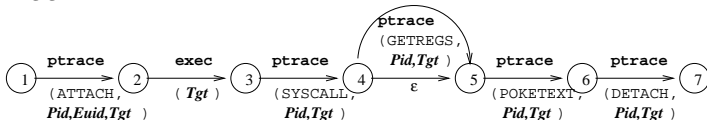
Misc

Architecture

Way Beyond

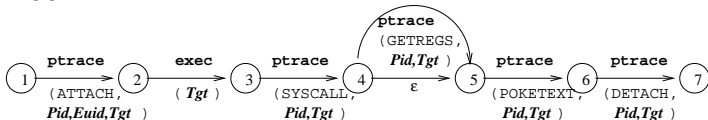
The Attack Signature

- We can count on the system **logging** important events.
Here we count on the SNARE kernel module.
We may also interface to the `syslog` facility.
- ORCHIDS will now try to find **patterns** among these logged events:



The Attack Signature

- We can count on the system **logging** important events.
Here we count on the SNARE kernel module.
We may also interface to the `syslog` facility.
- ORCHIDS will now try to find **patterns** among these logged events:



- Note that just detecting calls `ptrace` is **not** enough: this is used in everyday debugging activities, and is not indicative of an attack by itself.

A Monitor... Yes, Almost

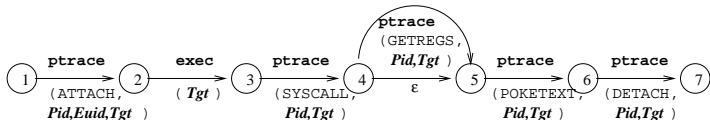
ORCHIDS

Laboratoire
Spécification
et Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain



will try to match a **subsequence** of all events, e.g.,

$A, \text{ptrace}(\text{ATTACH}, \dots), B, A, \text{exec}(\dots), \text{ptrace}(\text{SYSCALL}, \dots), A, A, B,$
 $\text{ptrace}(\text{GETREGS}, \dots), B, B, A, \text{ptrace}(\text{POKETEXT}, \dots), A,$
 $\text{ptrace}(\text{DETACH}, \dots), B, A, \dots$

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

A Monitor... Yes, Almost

ORCHIDS

Laboratoire
Specification
Verification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

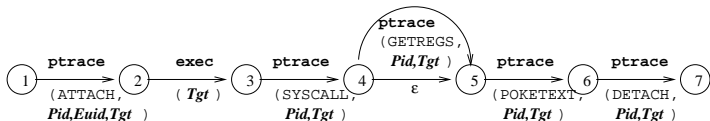
Beyond

Conclusion

Misc

Architecture

Way Beyond



will try to match a **subsequence** of all events, e.g.,

$A, \text{ptrace}(\text{ATTACH}, \dots), B, A, \text{exec}(\dots), \text{ptrace}(\text{SYSCALL}, \dots), A, A, B,$
 $\text{ptrace}(\text{GETREGS}, \dots), B, B, A, \text{ptrace}(\text{POKETEXT}, \dots), A,$
 $\text{ptrace}(\text{DETACH}, \dots), B, A, \dots$

Corollary: There is no unique matching subsequence.

A Monitor... Yes, Almost

ORCHIDS

Laboratoire
Specification
Verification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

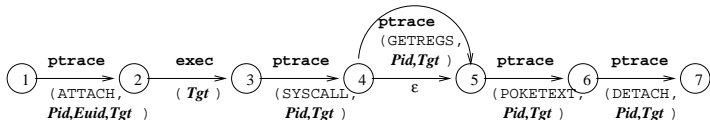
Beyond

Conclusion

Misc

Architecture

Way Beyond



will try to match a **subsequence** of all events, e.g.,

$A, \text{ptrace}(\text{ATTACH}, \dots), B, A, \text{exec}(\dots), \text{ptrace}(\text{SYSCALL}, \dots), A, A, B,$
 $\text{ptrace}(\text{GETREGS}, \dots), B, B, A, \text{ptrace}(\text{POKETEXT}, \dots), A,$
 $\text{ptrace}(\text{DETACH}, \dots), B, A, \dots$

Corollary: There is no unique matching subsequence.

We shall report those that are most informative (“shortest runs”, see later).

Demo

Julien Olivain



Now let's see Orchids in action.

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Demo — Reacting to an Intrusion

ORCHIDS

ORC ST
Laboratoire
Specification
Verification



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Julien Olivain



Jean, did you show them that
Orchids *killed* the offending
user's account?
Did you explain them why?

Demo — Reacting to an Intrusion

Julien Olivain



Jean, did you show them that Orchids killed the offending user's account?
Did you explain them why?

Sure, Julien:

- ▶ The attacker may have left a **backdoor** in the system, allowing him to reenter at will.
We should prevent him from using it later.
- ▶ Also, Orchids produces **reports!**
Here, tracks attacker's achievements.

ORCHIDS

SEC (SI) Laboratoire Specification Verification



Jean Goubault-Larrecq,
Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Demo — Escaping Masking Attacks

Julien Olivain



Jean, have you shown them that Orchids was not fooled by *masking* attacks?

ORCHIDS

ORC
ST
Laboratoire
Specification
Verification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Demo — Escaping Masking Attacks

ORCHIDS

SEC (ST) Laboratoire Specification Verification



Julien Olivain



Jean, have you shown them that Orchids was not fooled by *masking* attacks?

Oh yes. The point is that the attacker may attempt to **drown** the intruder detection system under many similar events.

Goal: attempt to escape detection.

Let's see how Orchids fares **under pressure** : let's generate zillions of benign calls to **ptrace**.

Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo Under the Hood

Beyond

Conclusion

Misc Architecture Way Beyond

How Does ORCHIDS Detect It?

ORCHIDS

ORCHIDS
Laboratoire
Spécification
Vérification



INRIA

Jean Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

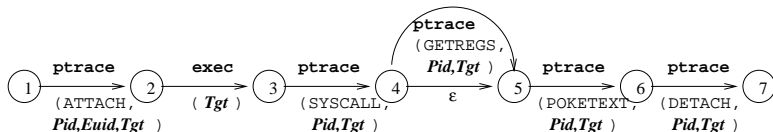
Beyond

Conclusion

Misc

Architecture

Way Beyond



Imagine the following flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)
```

```
ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)
```

(What to come, in a nutshell: the ORCHIDS engine will basically simulate a form of alternating automata, with additional optimizations gotten by abstract interpretation of the formulae.)

Detecting the Attack

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)      ptrace (SYSCALL, pid=100, 101)
ptrace (ATTACH, pid=57, euid=500, 58)           ptrace (GETREGS, pid=100, 101)
ptrace (ATTACH, pid=100, euid=500, 101)         ptrace (POKETEXT, pid=100, 101)
exec (prog="modprobe", pid=101)                ptrace (POKETEXT, pid=100, 101)
ptrace (ATTACH, pid=100, euid=500, 101)         ptrace (POKETEXT, pid=100, 101)
exit (pid=58)                                   ptrace (DETACH, pid=100, 101)
```

Initially, ORCHIDS has no active thread.

Detecting the Attack

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)
ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)
```

The signature contains no `open` event: skip.

Detecting the Attack

ORCHIDS

Laboratoire
Spécification
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

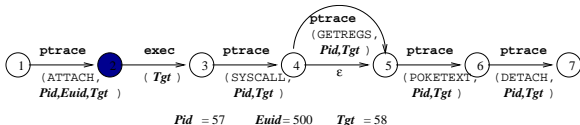
Architecture

Way Beyond

Flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)
```

```
ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)
```



Detecting the Attack

ORCHIDS

Laboratoire
Spécification
Vérification



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

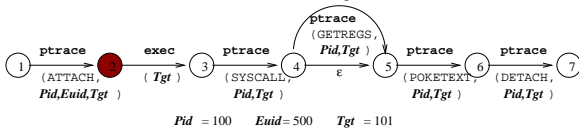
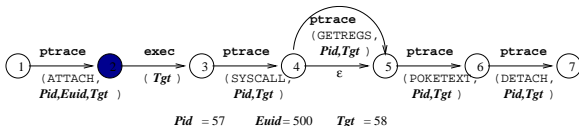
Misc

Architecture
Way Beyond

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



Spawn thread:
avoid **masking attacks**.

Detecting the Attack

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

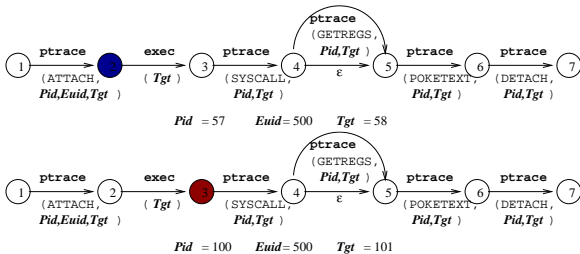
Architecture

Way Beyond

Flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)
```

```
ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)
```



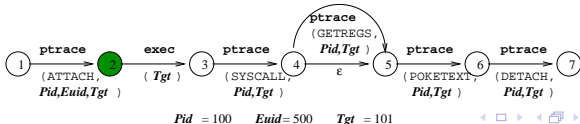
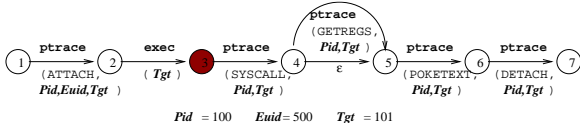
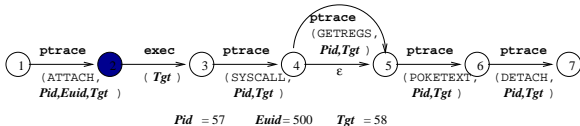
No need to spawn thread: would violate **shortest runs.**

Detecting the Attack

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



ORCHIDS

Laboratoire
Spécification
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

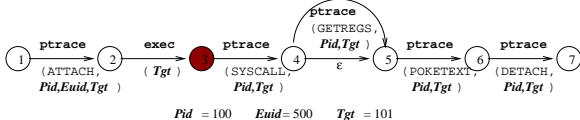
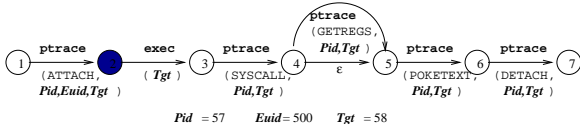
Architecture
Way Beyond

Detecting the Attack

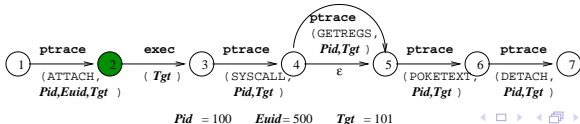
Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



Irrelevant event exit

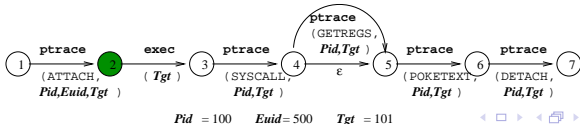
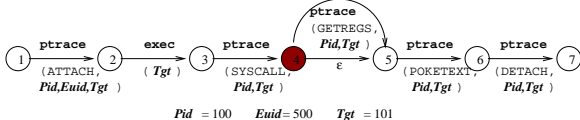
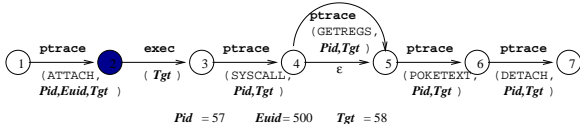


Detecting the Attack

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)

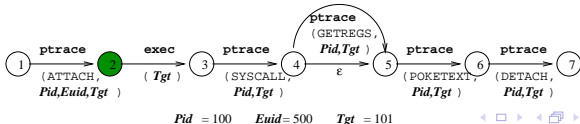
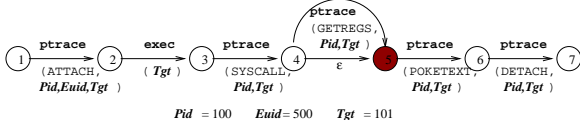
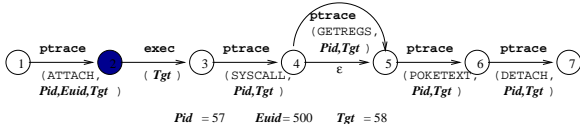


Detecting the Attack

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



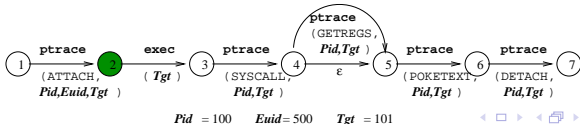
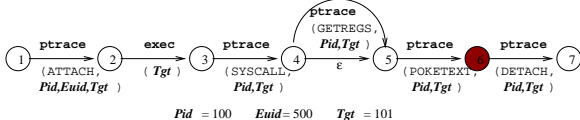
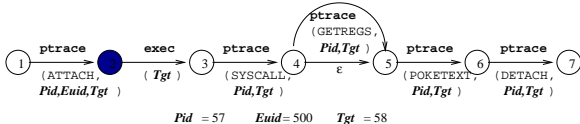
Keep run
ATTACH-exec-
SYSCALL-**GETREGS**
for reporting.

Detecting the Attack

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



Do not spawn thread
(shortest runs again)
(prefer A- over -A-, -A
on trace AAA).
Formally: keep
lexicographically
smallest
sequences of
event numbers.

ORCHIDS

Laboratoire
Spécification
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

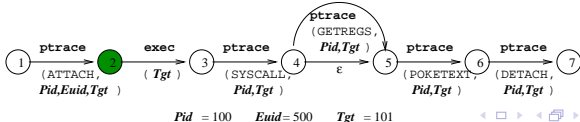
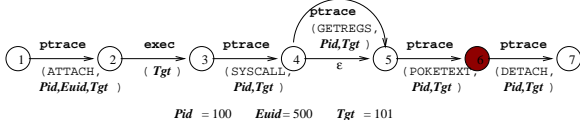
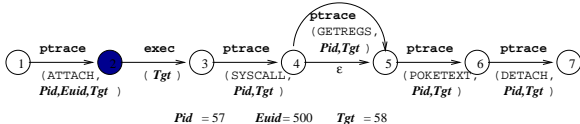
Misc
Architecture
Way Beyond

Detecting the Attack

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



Do not spawn thread (shortest runs again) (prefer A- over -A-, -A on trace AAA). Formally: keep lexicographically smallest sequences of event numbers.



Jean Goubault-Larrecq, Julien Olivain

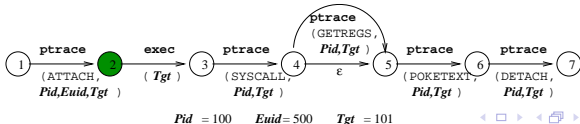
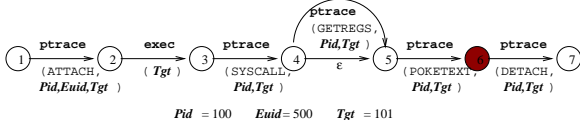
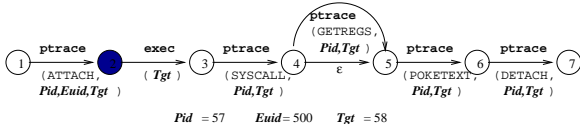
- Introduction
- Issues
- Example Attack
- Running ORCHIDS
- Demo
- Under the Hood
- Beyond
- Conclusion
- Misc
- Architecture
- Way Beyond

Detecting the Attack

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



Do not spawn thread (shortest runs again) (prefer A- over -A-, -A on trace AAA). Formally: keep lexicographically smallest sequences of event numbers.

ORCHIDS

Laboratoire
Spécification
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

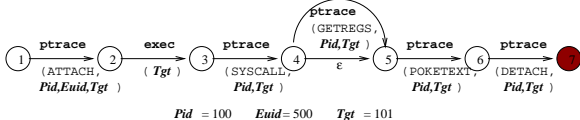
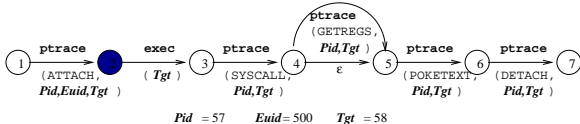
Way Beyond

Detecting the Attack

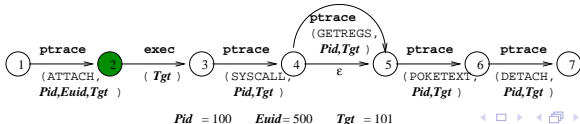
Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



Alert.



Shortest Runs (1)

ORCHIDS

ORCHIDS
Laboratoire
Specification
Verification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

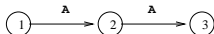
Conclusion

Misc

Architecture

Way Beyond

Threads can usually match several subsequences. E.g.



would match (starting from first element):

A A A A A A A A

: 1, 2

Shortest Runs (1)

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

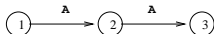
Conclusion

Misc

Architecture

Way Beyond

Threads can usually match several subsequences. E.g.



would match (starting from first element):

A A A A A A A A

: 1, 3

Shortest Runs (1)

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

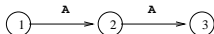
Conclusion

Misc

Architecture

Way Beyond

Threads can usually match several subsequences. E.g.



would match (starting from first element):

A A A A A A A A

: 1,4

Shortest Runs (1)

ORCHIDS

ORCHIDS
Laboratoire
Specification
Verification



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

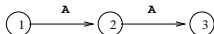
Conclusion

Misc

Architecture

Way Beyond

Threads can usually match several subsequences. E.g.



would match (starting from first element):

A A A A A A A A

: 1,5

Shortest Runs (1)

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

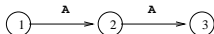
Conclusion

Misc

Architecture

Way Beyond

Threads can usually match several subsequences. E.g.



would match (starting from first element):

A A A A A A A A A A

: 1, 6

Shortest Runs (1)

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

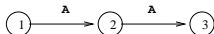
Conclusion

Misc

Architecture

Way Beyond

Threads can usually match several subsequences. E.g.



would match (starting from first element):

A A A A A A A A A

: 1, 7

Shortest Runs (1)

ORCHIDS

ORCHIDS
Laboratoire
Specification
Verification



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

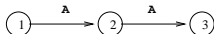
Conclusion

Misc

Architecture

Way Beyond

Threads can usually match several subsequences. E.g.



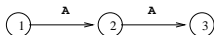
would match (starting from first element):

A A A A A A A A

: 1,8

Shortest Runs (1)

Threads can usually match several subsequences. E.g.



would match (starting from first element):

A A A A A A A A

: 1, 8

Definition

A **run** is an increasing sequence of integer positions
 $i_1 < i_2 < \dots < i_k, k \geq 1$.

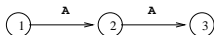
We compare runs by $(i_1, i_2, \dots, i_k) \preceq (j_1, j_2, \dots, j_\ell)$ iff:

- $i_1 = j_1 \dots$ (they start at the same place);
- $i_k \leq j_\ell \dots$ (the shorter one ends earlier)
- and ...

We only return \preceq -minimal (“shortest”) runs.

Shortest Runs (1)

Threads can usually match several subsequences. E.g.



would match (starting from first element):

A A A A A A A A

: 1, 2

Definition

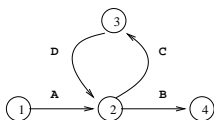
A **run** is an increasing sequence of integer positions
 $i_1 < i_2 < \dots < i_k, k \geq 1$.

We compare runs by $(i_1, i_2, \dots, i_k) \preceq (j_1, j_2, \dots, j_\ell)$ iff:

- $i_1 = j_1 \dots$ (they start at the same place);
- $i_k \leq j_\ell \dots$ (the shorter one ends earlier)
- and ...

We only return \preceq -minimal (“shortest”) runs.

Shortest Runs (2)



would match (starting from first element):

A C D C D C D C B

: 1, 9

Definition

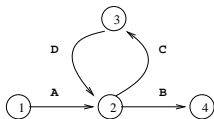
A **run** is an increasing sequence of integer positions

$$i_1 < i_2 < \dots < i_k, k \geq 1.$$

We compare runs by $(i_1, i_2, \dots, i_k) \preceq (j_1, j_2, \dots, j_l)$ iff:

- $i_1 = j_1 \dots$ (they start at the same place);
- $i_k \leq j_l \dots$ (the shorter one ends earlier)
- and ...

Shortest Runs (2)



would match (starting from first element):

A C D C D C D C B

: 1, 2, 3, 9

Definition

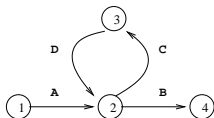
A **run** is an increasing sequence of integer positions

$$i_1 < i_2 < \dots < i_k, k \geq 1.$$

We compare runs by $(i_1, i_2, \dots, i_k) \preceq (j_1, j_2, \dots, j_l)$ iff:

- $i_1 = j_1 \dots$ (they start at the same place);
- $i_k \leq j_l \dots$ (the shorter one ends earlier)
- and ...

Shortest Runs (2)



would match (starting from first element):

A C D C D C D C B

: 1, 4, 5, 9

Definition

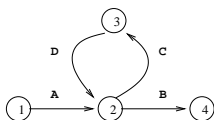
A **run** is an increasing sequence of integer positions

$$i_1 < i_2 < \dots < i_k, k \geq 1.$$

We compare runs by $(i_1, i_2, \dots, i_k) \preceq (j_1, j_2, \dots, j_l)$ iff:

- $i_1 = j_1 \dots$ (they start at the same place);
- $i_k \leq j_l \dots$ (the shorter one ends earlier)
- and ...

Shortest Runs (2)



would match (starting from first element):

A C D C D C D C B

: 1, 4, 7, 9

Definition

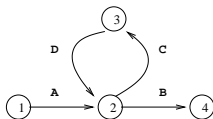
A **run** is an increasing sequence of integer positions

$$i_1 < i_2 < \dots < i_k, k \geq 1.$$

We compare runs by $(i_1, i_2, \dots, i_k) \preceq (j_1, j_2, \dots, j_l)$ iff:

- $i_1 = j_1 \dots$ (they start at the same place);
- $i_k \leq j_l \dots$ (the shorter one ends earlier)
- and ...

Shortest Runs (2)



would match (starting from first element):

A C D C D C D C B

: 1, 2, 3, 4, 5, 6, 7, 9

Definition

A **run** is an increasing sequence of integer positions

$$i_1 < i_2 < \dots < i_k, k \geq 1.$$

We compare runs by $(i_1, i_2, \dots, i_k) \preceq (j_1, j_2, \dots, j_\ell)$ iff:

- $i_1 = j_1 \dots$ (they start at the same place);
- $i_k \leq j_\ell \dots$ (the shorter one ends earlier)
- and $(i_1, i_2, \dots, i_k) \leq_{lex} (j_1, j_2, \dots, j_\ell)$.

ORCHIDS

Laboratoire
Specification
Verification

ENS
CYRS

INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

Shortest Runs

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Shortest runs are exploited in two ways:

- First, we keep threads **sorted** in such a way that the first thread that reaches the alert state is the one matching the shortest run with a given i_1 — we kill the others.

Shortest Runs

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Shortest runs are exploited in two ways:

- First, we keep threads **sorted** in such a way that the first thread that reaches the alert state is the one matching the shortest run with a given i_1 — we kill the others.
- More important: we **don't create** threads that may match some runs, but never the shortest ones.

Not Creating Useless Threads

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)
ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)
```

Initially, ORCHIDS has no active thread.

Not Creating Useless Threads

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

Flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)
ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)
```

The signature contains no `open` event: skip.

Not Creating Useless Threads

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

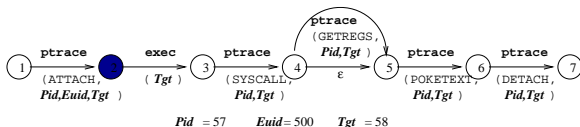
Architecture

Way Beyond

Flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)
```

```
ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)
```



Not Creating Useless Threads

ORCHIDS

Laboratoire
Spécification
Vérification

ENS
CLRS

INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

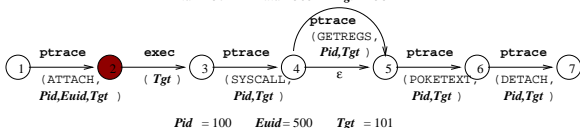
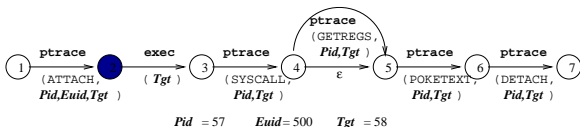
Misc

Architecture
Way Beyond

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



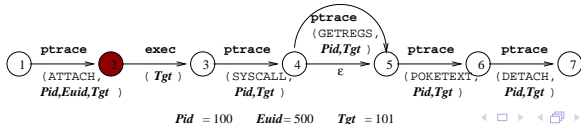
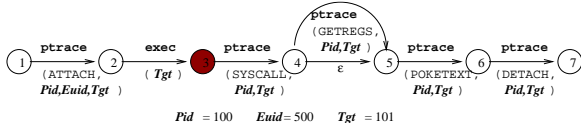
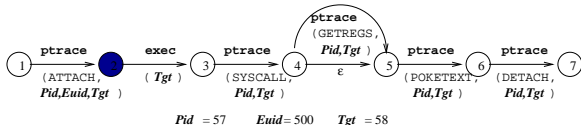
Spawn thread:
avoid **masking attacks**.

Not Creating Useless Threads

Flow of events:

open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)

ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)



ORCHIDS

Laboratoire
Spécification
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

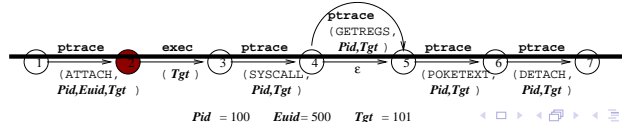
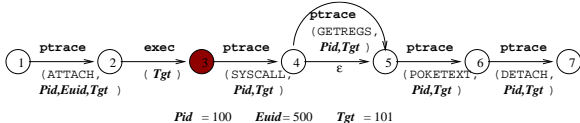
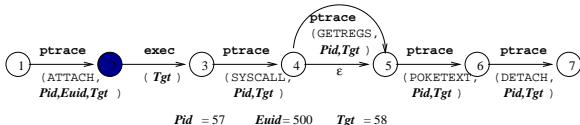
Way Beyond

Not Creating Useless Threads

Flow of events:

```
open ("/etc/passwd", "r", pid=58, euid=500)
ptrace (ATTACH, pid=57, euid=500, 58)
ptrace (ATTACH, pid=100, euid=500, 101)
exec (prog="modprobe", pid=101)
ptrace (ATTACH, pid=100, euid=500, 101)
exit (pid=58)
```

```
ptrace (SYSCALL, pid=100, 101)
ptrace (GETREGS, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (POKETEXT, pid=100, 101)
ptrace (DETACH, pid=100, 101)
```



No need to
spawn thread:
would violate
shortest runs.

ORCHIDS

Laboratoire
Spécification
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Outline

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

- 1 Some Security Issues
- 2 Example: The `ptrace` Attack
- 3 Detecting The Attack, Using ORCHIDS
 - Demo
 - Under the Hood
- 4 Beyond: Additional Features, Further Attacks
- 5 Conclusion
- 6 Other Things That Cannot Fit In The Talk
 - The Architecture of ORCHIDS
 - Way Beyond

Guards

ORCHIDS



Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo

Under the Hood

Beyond

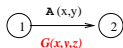
Conclusion

Misc

Architecture

Way Beyond

Each transition can be labeled with a *guard* G , e.g.,



These depend on variables that may be:

- *flexible*: i.e., may vary through time;
- or *rigid*: once x gets a value, it keeps it forever.

Monotonic Variables

Some variables are labeled **monotonic**, and are assumed to only increase through time.

E.g., **date** fields.

Observation

If the guard $G(\vec{x}, \vec{y}, \vec{z})$ is:

- *antitone* in its monotonic variables \vec{x} ,
(if variable increases, then G can only go from true to false, or remain unchanged.)
- *independent* of its non-monotonic flexible variables \vec{y} ,

and if $G(\vec{x}, \vec{y}, \vec{z})$ is *false* at event i , then it will remain false at all events $j \geq i$.

Hence we can safely kill threads waiting on such guards $G(\vec{x}, \vec{y}, \vec{z})$.

Note: Elegant generalization of **timeouts** (" $t < 3600$ ").

ORCHIDS

ORCHIDS
Laboratoire
Specification
Verification



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Cuts

Cuts (à la Prolog): generalize shortest runs.
Enable one to remove threads thought to be irrelevant.

ORCHIDS

Laboratoire
Specification
Verification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Cuts

Cuts (à la Prolog): generalize shortest runs.
Enable one to remove threads thought to be irrelevant.

Green cuts: Preserve the *no-masking* property:

Each attack (family of runs) \Rightarrow at least one alert is reported.

Monotonic variables are an example.

Shortest runs are another example, with an extra optimality property: ≤ 1 alert is reported for any given i_1 .

ORCHIDS

Laboratoire
Specification
Verification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Cuts

Cuts (à la Prolog): generalize shortest runs.
Enable one to remove threads thought to be irrelevant.

Green cuts: Preserve the *no-masking* property:

Each attack (family of runs) \Rightarrow at least one alert is reported.

Monotonic variables are an example.

Shortest runs are another example, with an extra optimality property: ≤ 1 alert is reported for any given i_1 .

Red cuts: Forget attacks.

- Avoid congestion, avert attacks on the IDS.
- Simulate **monitors**. (case where no thread is spawned dynamically.)
- **Intermediate** reports. (emit them, kill threads having emitted none, then proceed.)
- Implement the `without` operator (check that no event satisfying F occurs while waiting for G).

(e.g., kill threads when monitored process exits.)

ORCHIDS

Laboratoire
Specification
Verification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond



Not Just One Attack, But Families

ORCHIDS



Julien Olivain



Jean, I'm afraid you did not insist on the fact that you could catch whole *families* of attacks by just one ORCHIDS signature rule. Have you demonstrated the `do_brk` attack, by the way?

Jean Goubault-Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

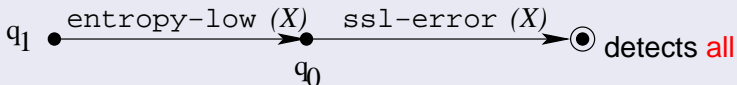
Conclusion

Misc
Architecture
Way Beyond

Detect Families of Attacks

Signatures can be made to match not just one attack, rather whole **families**.

The Entropy Checker



buffer overflow attacks on crypto protocols (ssh1, ssh2, https, ldaps, ...).

- By the way, this is a network attack. ORCHIDS is not limited to system attacks.
- By the way, it uses the Net-Entropy sensor to detect entropy anomalies in input flow.

ORCHIDS

ORCHIDS
Laboratoire
Specification
Verification

ORCHIDS
ORCHIDS

INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

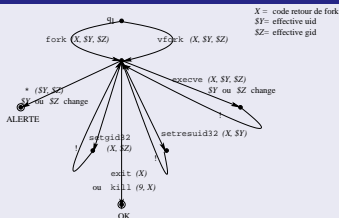
Misc

Architecture
Way Beyond

Detect Families of Attacks

Signatures can be made to match not just one attack, rather whole **families**.

The Pid Tracker



detects all attacks in the style of

`do_brk`, `mmap`, `munmap`, `mremap`, etc. [MortonStarzetz'03].

- Principle: detect that some user gained root privileges without using the authorized `set*id` mechanism, **whichever** way he actually managed to do so.
- Technically, a form of dynamically-spawned monitors.

The Attacks on `do_brk`, `mmap`, `munmap`, ... [MortonStarzetz03]

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

Beyond

Conclusion

Misc

Architecture

Way Beyond

One of the most **serious** attacks ever actually used.

- Crackers used it to bog down the Savannah (GNU) and Debian servers in 2004. Downtime: **several weeks**.

The Attacks on `do_brk`, `mmap`, `munmap`, ... [MortonStarzetz03]

ORCHIDS



Jean Goubault-Larrecq,
Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

One of the most **serious** attacks ever actually used.

- Crackers used it to bog down the Savannah (GNU) and Debian servers in 2004. Downtime: **several weeks**.
- **Vicious** attacks: give rise to **no** event at all.

... except a flurry of calls to `do_brk` (but this is what `malloc` calls!)

... except a flurry of `SIGSEGV` signals (not logged by `SNARE`!)

- Principle: rewrite the information the kernel keeps on our (user) process by mapping the kernel into the address space of the process (!).

Outline

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

- 1 Some Security Issues
- 2 Example: The `ptrace` Attack
- 3 Detecting The Attack, Using ORCHIDS
 - Demo
 - Under the Hood
- 4 Beyond: Additional Features, Further Attacks
- 5 Conclusion
- 6 Other Things That Cannot Fit In The Talk
 - The Architecture of ORCHIDS
 - Way Beyond

Conclusion

ORCHIDS, an efficient **on-line, real-time** intrusion prevention system.

- Handles real, recent, sophisticated attacks;
- Produces detailed reports, runs countermeasures;
i.e., emergency measures until the sys. admin. reacts.
- Multi-sensor, multi-port, multi-event. . . multi-whatever.
- . . . and based on technology very much related to RV!

ORCHIDS

ORCHIDS
Laboratoire
Spécification
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

Conclusion

ORCHIDS, an efficient **on-line, real-time** intrusion prevention system.

- Handles real, recent, sophisticated attacks;
- Produces detailed reports, runs countermeasures;
i.e., emergency measures until the sys. admin. reacts.
- Multi-sensor, multi-port, multi-event. . . multi-whatever.
- . . . and based on technology very much related to RV!

Contact:

Julien Olivain

olivain@lsv.ens-cachan.fr



Jean Goubault-Larrecq

goubault@lsv.ens-cachan.fr



ORCHIDS

Laboratoire
Specification
Verification



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

Outline

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

- 1 Some Security Issues
- 2 Example: The `ptrace` Attack
- 3 Detecting The Attack, Using ORCHIDS
 - Demo
 - Under the Hood
- 4 Beyond: Additional Features, Further Attacks
- 5 Conclusion
- 6 Other Things That Cannot Fit In The Talk
 - The Architecture of ORCHIDS
 - Way Beyond

The mod_ssl Attack [McDonald03]

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Exploits a buffer overflow in the OpenSSL code for Apache, implementing SSL v.2.

Effect: **remote** exploit, obtaining a remote shell.

The mod_ssl Attack [McDonald03]

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Exploits a buffer overflow in the OpenSSL code for Apache, implementing SSL v.2.

Effect: **remote** exploit, obtaining a remote shell.

- Extremely complex attack.
- Exploits several Apache threads to get vulnerability information.
- Transmits vulnerability information through **encrypted** SSL channel.

The SSL v2 Handshake Protocol

ORCHIDS

Laboratoire
de
Sécurité
et
de
Certification



INRIA

Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

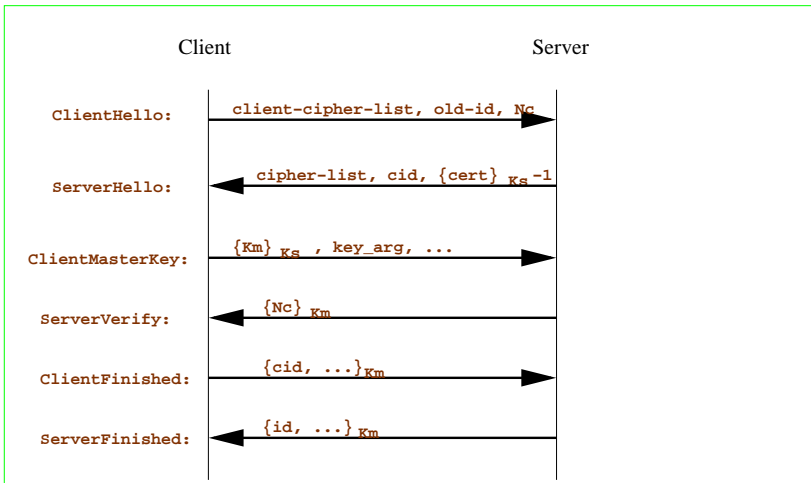
Demo Under the Hood

Beyond

Conclusion

Misc

Architecture Way Beyond



An (Important) Detail of Implementation in OpenSSL

ORCHIDS

Laboratoire
Spécification
Vérification



INRIA

Jean Goubault-Larrecq, Julien Olivain

Introduction

Issues

Example Attack

Running ORCHIDS

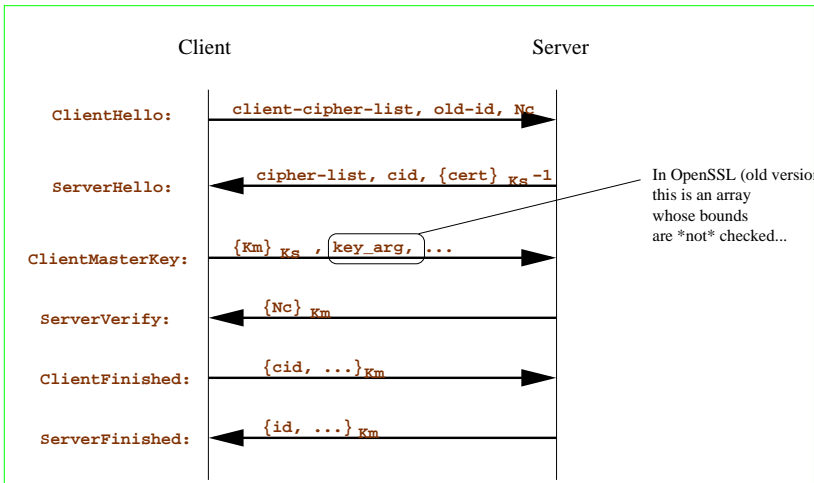
Demo Under the Hood

Beyond

Conclusion

Misc

Architecture Way Beyond




An Attack on OpenSSL

When the server receives ClientMasterKey, it copies it into:

```
typedef struct ssl_session_st
{
    int ssl_version;
    unsigned int key_arg_length;
    unsigned char key_arg[SSL_MAX_KEY_ARG_LENGTH];
    int master_key_length;
    unsigned char master_key[SSL_MAX_MASTER_KEY_LENGTH];
    [...]
    struct ssl_session_st *prev, *next;
} SSL_SESSION;
```

Vulnerable: let's
stuff it!



ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

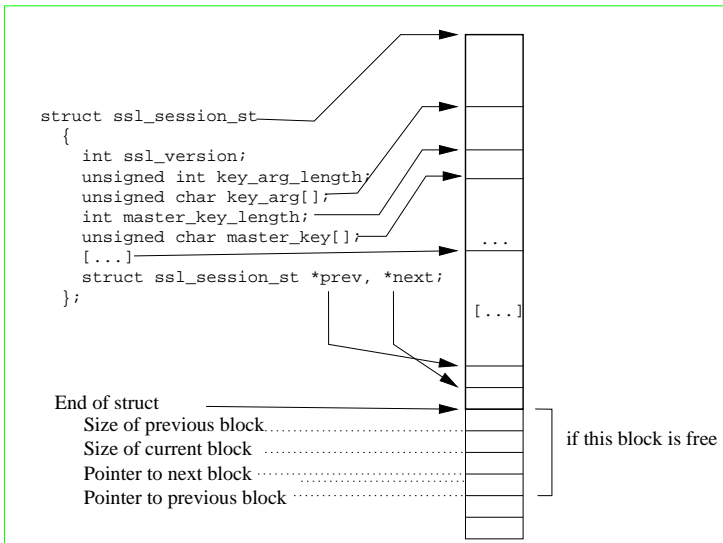
Demo
Under the Hood

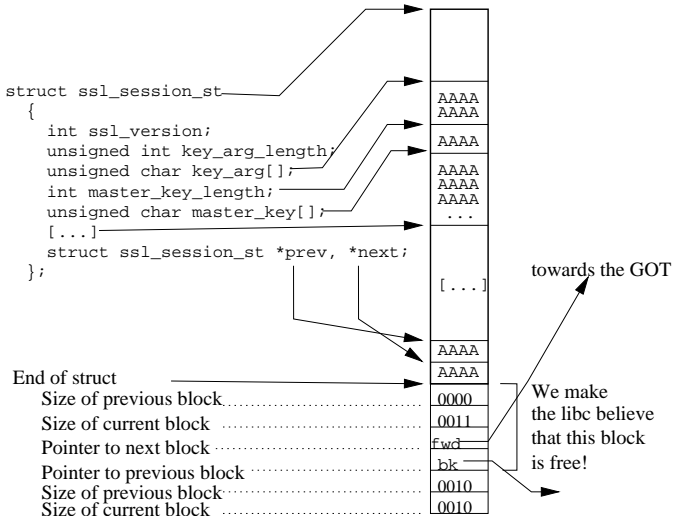
Beyond

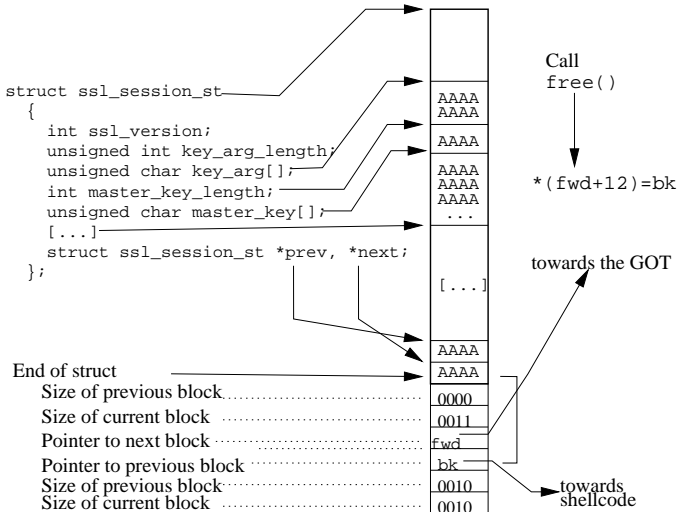
Conclusion

Misc

Architecture
Way Beyond







The rest of the attack, in short

- To retrieve the address of the shellcode, have the server retransmit all needed information by writing into the `session_id` field: we get the information, encrypted, in the `ServerFinished` message.
- We now know at which addresses the server works.
- Now replay a similar attack in a second SSL session to really execute the shellcode.
- The server now serves a connection to an `apache` or `nobody` shell through HTTP. (Now play a user-to-root attack. . .)

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Describing the attack

ORCHIDS

ORCHIDS
Laboratoire
Specification
Verification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo

Under the Hood

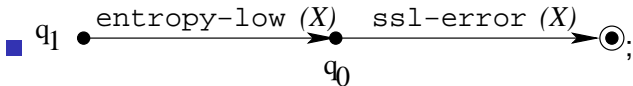
Beyond

Conclusion

Misc

Architecture

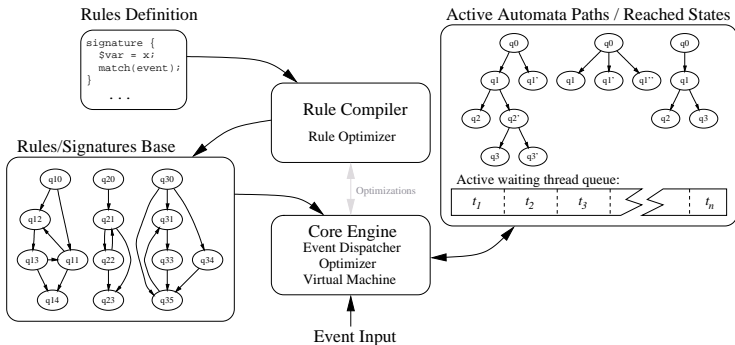
Way Beyond



- Note: use an **entropy** input module (to be published);
- Detects **any** attack on SSL where plain texts are served instead where we expect ciphertexts.

A Bird's Eye View

- Efficient implementation, through compilation to **bytecode trees**.



ORCHIDS

Laboratoire
Spécification
Vérification

ENS
ORS

INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

A Modular Architecture

ORCHIDS

Laboratoire
de
Sécurité
et
de
Vérification



INRIA

Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

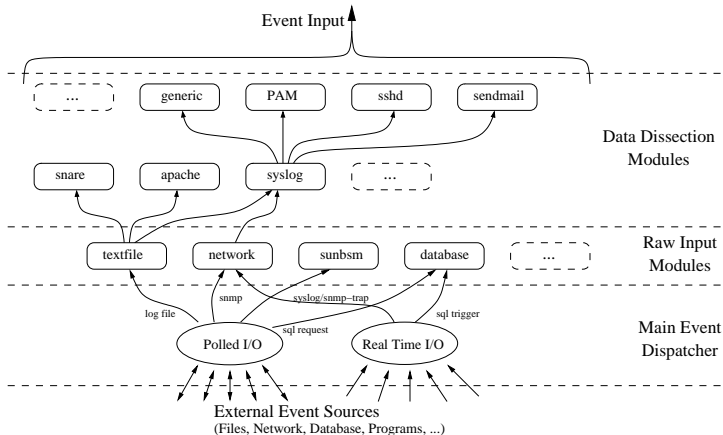
Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond



Input Sources

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc

Architecture
Way Beyond

Multi-sensor:

- **System** sensors:
 - SNARE, `syslog` (Unix);
 - MS EVT (Windows);
- **Network** and equipment sensors:
 - Cisco logs;
 - SNMP sensors;
 - Linux NetFilter.
 - ...
- Meta-sensors: e.g., SNORT used as a sensor.
- Filters: e.g., NetEntropy entropy tester.
- ...

Input Sources

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture

Way Beyond

Multi-sensor:

- **System** sensors:
 - SNARE, syslog (Unix);
 - MS EVT (Windows);
- **Network** and equipment sensors:
 - Cisco logs;
 - SNMP sensors;
 - Linux NetFilter.
 - ...
- Meta-sensors: e.g., SNORT used as a sensor.
- Filters: e.g., NetEntropy entropy tester.
- ...

Multi-port: reads from UDP connections, SNMP connections, log files.

Information Correlation

Use an embedded Prolog interpreter. Applications include:

Keep set of attacks that have already succeeded.

- Maintain **Black Lists**.
- Realize that user A has succeeded in the past in mounting an attack on machine M giving him a user account, allowing him to try and launch a second attack on M in the hope of gaining root privileges.

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture
Way Beyond

Information Correlation

ORCHIDS



Jean
Goubault-
Larrecq,
Julien Olivain

Introduction

Issues

Example
Attack

Running
ORCHIDS

Demo
Under the Hood

Beyond

Conclusion

Misc
Architecture

Way Beyond

Use an embedded Prolog interpreter. Applications include:

Reason about network topology.

- realize that `127.0.0.1` and `localhost` are the same machine (as in the M2D2 model [MMDD03]);
- realize that two machines *A* and *B* are neighbors and may have cooperated in mounting an attack.