

A decorative graphic at the top of the slide features a green sphere on the left and three overlapping semi-circles in light blue, light red, and light yellow on the right.

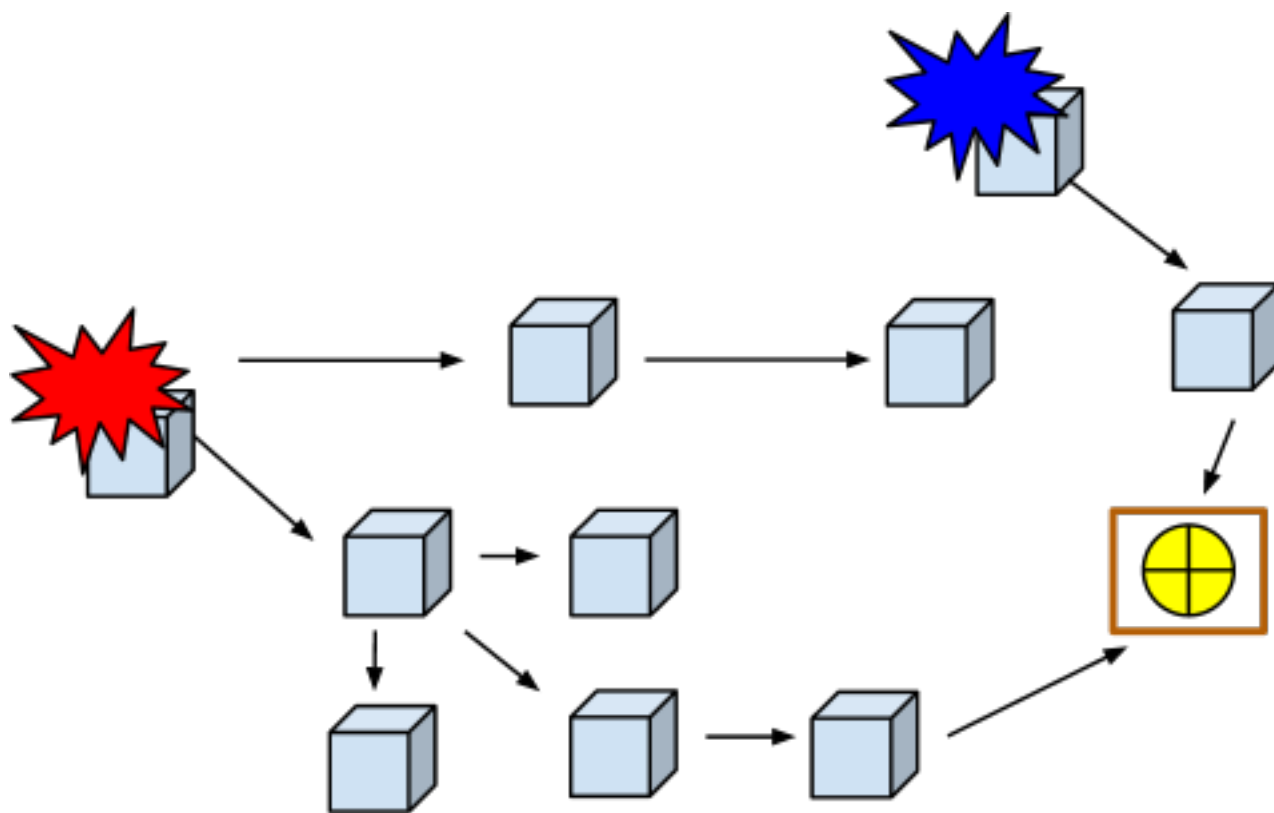
Distributed Forensics and Incident Response in the Enterprise

M.I. Cohen, D. Bilby, G. Caronni
Google Switzerland

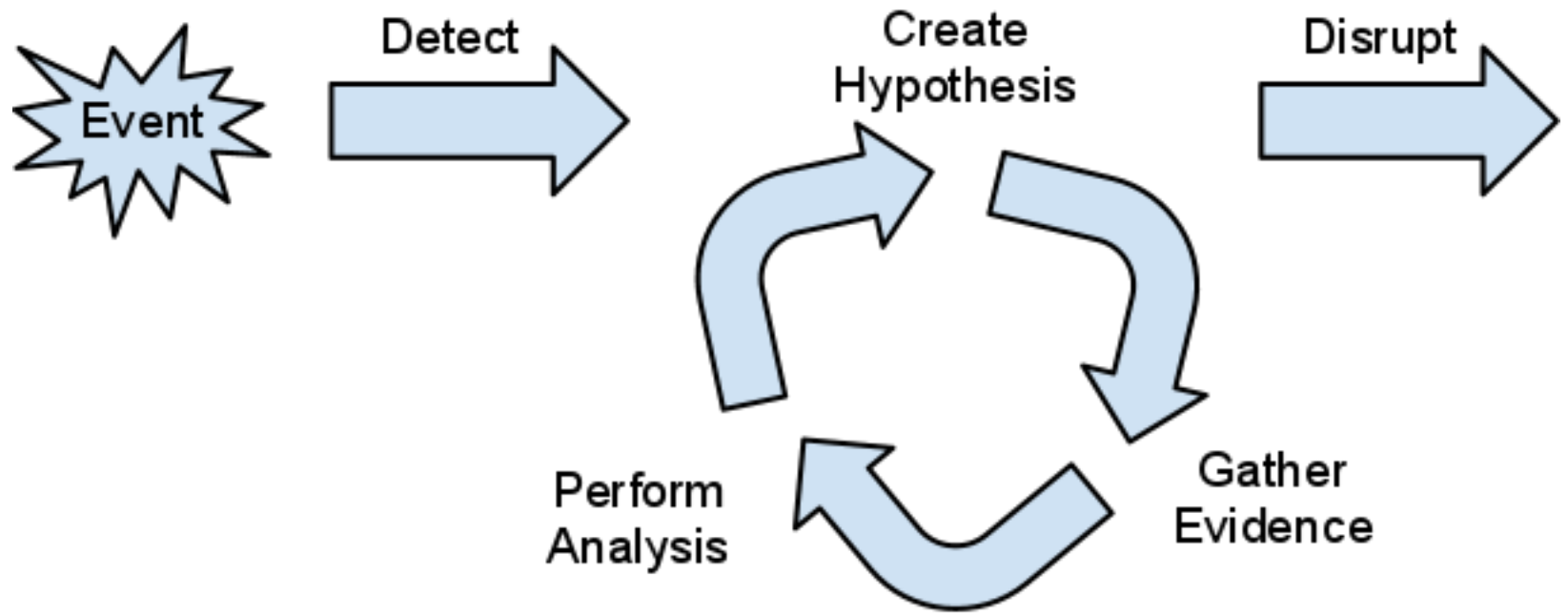


Investigation Goals

- What is the scope of the incident?
- What occurred on our systems/network?
- What was the entry point ?



The Response Cycle



- Hypothesize, Gather, Analyze
- Your time to execute the cycle determines your success in complex incidents
- Tools must reduce the duration of the cycle

The Challenges

- Distributed offices / timezones
- Scalability 2, 200, 20'000 machines
- System diversity
- Roaming / travelling users
- Limited number of skilled responders
- Intelligent attackers

Live Response - GRR

Incident Response is a Search Problem!

- You need an agent on the host
- You need to be able to access it over the Internet
- You need a system that scales
- You need to automate response

Wishlist:

"Find me any machine that has a wiped event log in the last 10 hours"

"Find me all machines with a driver signed by Realtek in % SYSTEMDIR% that contain the string "labarumY"

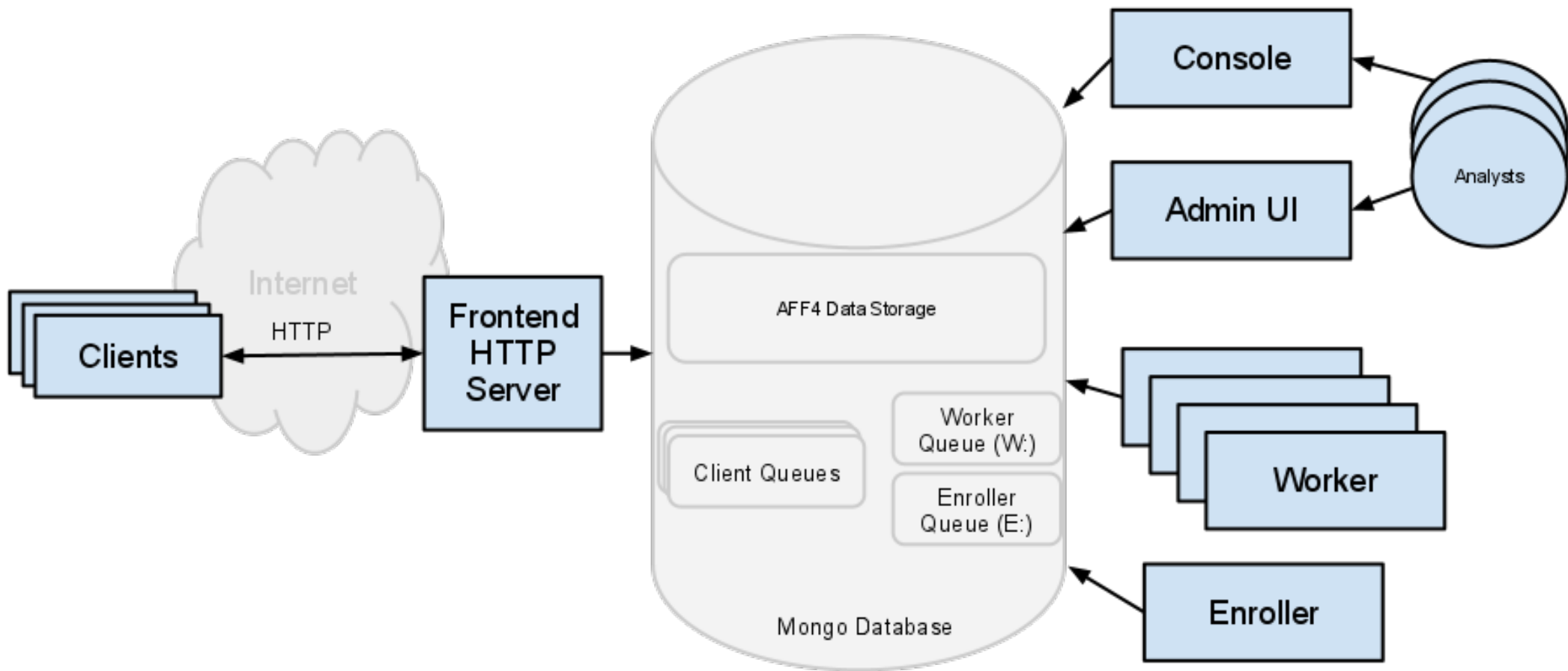


Not as easy as it sounds

- Scalable processing (1000's of machines)
- Data model
- Subverting live forensics
- Unreliable client connections
- Intelligent automation

- Security (Discussed in the paper)
- Privacy (Discussed in the paper)
- ...

GRR Overview



Not as easy as it sounds

- **Scaleable processing (1000's of machines)**
- Data model
- Subverting live forensics
- Unreliable client connections
- Intelligent automation

Scaleable Processing

- Traditional forensic model 1:1 server -> client won't scale
 - Need to move some work to the client
 - More workers to scale analysis
 - Messaging protocol and queueing
 - Client maybe offline
 - Offline analysis
 - Evidence collection scheduling
 - Suspendable flows - no resources used on server when client is not available.
- Large quantities of data:
 - Sharded database
 - Map Reduce



Not as easy as it sounds

- Scalable processing (1000's of machines)
- **Data model**
- Subverting live forensics
- Unreliable client connections
- Intelligent automation

Data Model

- GRR deals with a lot of data in a uniform way
 - Many hosts we want to access
 - Partial information about each host
 - Billions of objects
- This model is suitable for AFF4
 - AFF4 - one universal data model
 - Object Oriented
 - A way of organizing forensic information
 - Based on RDF or NoSQL models:
 - All objects have a unique ID (AFF4 URN).
 - Objects can be sharded across many servers
 - RDF allows cross referencing

Data Model: AFF4 Objects Examples

aff4:/C.91dcaaa12616af24

aff4:/C.91dcaaa12616af24/fs/os/c/Windows/
...Prefetch/DEFRAG.EXE-273F131E.pf

aff4:/C.91dcaaa34616af24/registry/
...HKEY_LOCAL_MACHINE/SOFTWARE/pidgin/Version

aff4:/C.91dcaaa34616af24/dev/PhysicalDrive0/2048
.../Windows/system32/Config/SOFTWARE/pidgin/Version

aff4:/C.91dcaaa34616af24/processes/1182/exe



Data Model: AFF4 Attributes

aff4:/C.91dcaaa12616af24 aff4:hostname "foobar"

aff4:/C.91dcaaa12616af24 aff4:os_version "5.1.2600"



Data Model: AFF4 Extensions

- Original AFF4 specifications were aimed at storing static forensic data in a forensic volume.
 - We do not need to create a AFF4 storage, but we use the AFF4 data model within the RDF data store
- Data is dynamic - a directory listing today is not the same as the same directory listing in the future.
 - All attributes now have an Age:
 - We still retain all the data, but sort by age
 - Incoherency is inherent in the entire system and must guide our reasoning
 - Age of data is important for historical diffs

Not as easy as it sounds

- Scalable processing (1000's of machines)
- Data model
- **Subverting live forensics**
- Unreliable client connections
- Intelligent automation

Subverting Live Forensics

- We can never fully protect the agent from subversion
- We can make it harder for the attacker:
 - Agent has multiple ways to access the same forensic data, e.g. through the APIs, forensic analysis of raw devices.
- What can an attacker do with our agent?
 - Hook OS APIs
 - Hook raw disk access
 - Replace our agent with one which reports "All is well"
 - Intercept and replay communications



Not as easy as it sounds

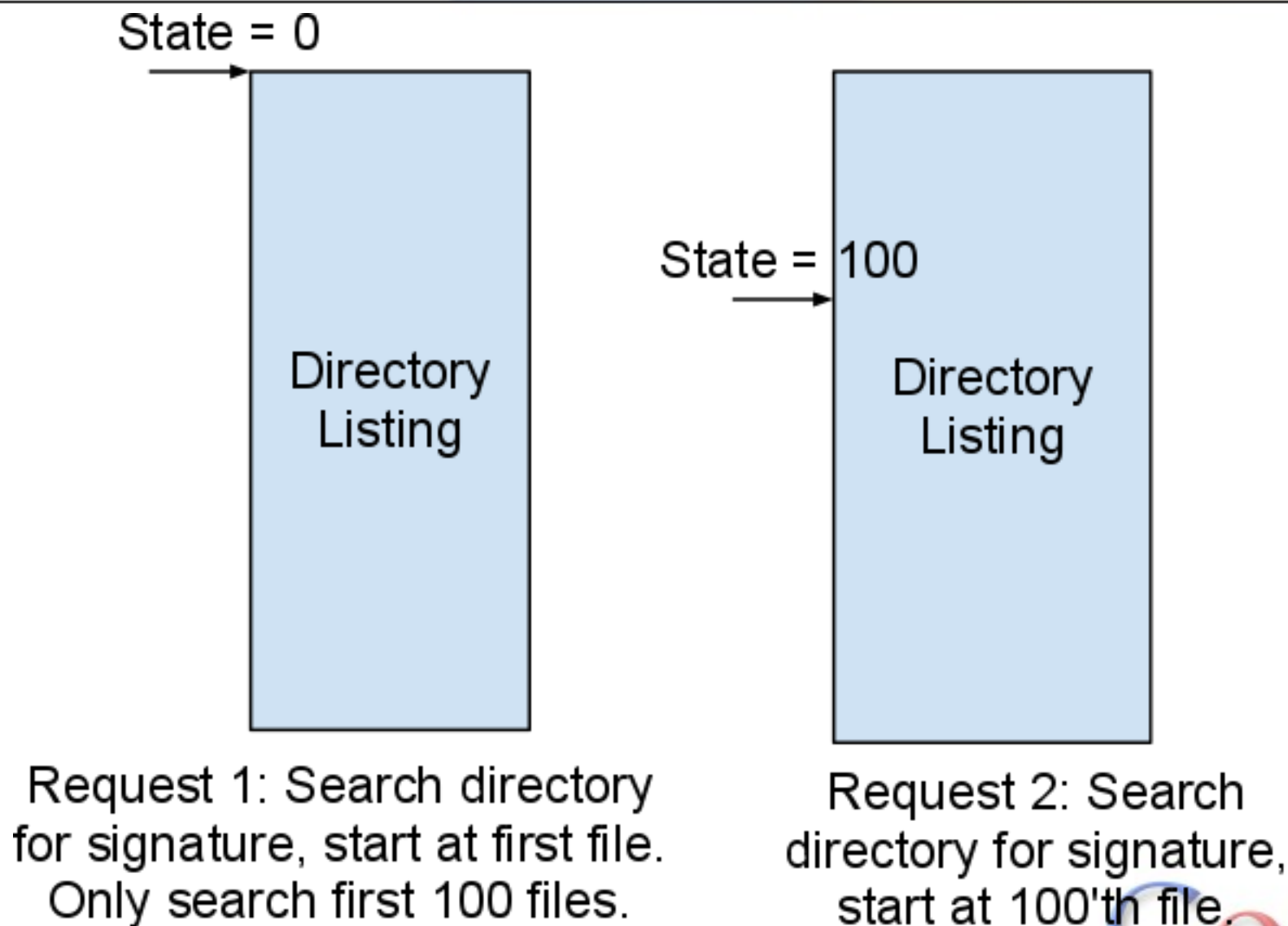
- Scalable processing (1000's of machines)
- Data model
- Subverting live forensics
- **Unreliable client connections**
- Intelligent automation

Unreliable Client Connections

- Client keeps very little state
 - Small footprint
 - Simplicity
- Some actions take a long time
 - e.g. Search the entire filesystem
- Big problem when machines disappear
- Solution: Iterated Flows
 - Client actions maintain internal state in a data structure
 - Data structure is periodically returned to the server flow
 - Automatic checkpoints progress



Example: Iterated Grep



Not as easy as it sounds

- Scalable processing (1000's of machines)
- Data model
- Subverting live forensics
- Unreliable client connections
- **Intelligent automation**

Intelligent Automation

- Automating analysis can help to reduce response time, and increase fleet coverage.
 - There is a *Foreman* flow which all clients call to periodically.
 - The Foreman distributes jobs to each client depending on certain criteria. For example:
 - All Windows 7 machines from a certain department run flow "CheckForBadMalware".
 - All systems with a running service named "srv.exe", retrieve their memory image.
- Automation can be initiated from other systems such as Security Event Manager (SEM) or Intrusion Detection System (IDS)



What We Have So Far

- Proof of Concept code
- Open Source Apache Licensed
- Basic automated analysis e.g. Get Browser History
- File system raw read e.g. \$MFT, deleted files
- Memory retrieval on Windows
- Web UI
- Windows, OSX, Linux clients
- Protobufs for serialization
- MongoDB NOSQL database backed
- Python + libraries
- Sleuthkit, PyTSK
- OpenSSL for encryption



Future Work

- Extend automated analysis
- We have great tools but no way to apply them against remote systems.
 - GRR is a framework to extend analysis to remote systems
- Extend existing tools to the enterprise:
 - volatility
 - regripper
 - log2timeline
- Privacy controls

Questions?



<http://code.google.com/p/grr>

Questions?

Michael Cohen
Darren Bilby
Germano Caronni
Andreas Moser
Joachim Metz

