Gibson:

3D Visualization and Modeling of Real Time Security Events

Dan Klinedinst
gibson3d.org
@dklinedinst

Who Am I?

- Security Researcher at Carnegie Mellon University
- Security of enterprise systems
- Primarily Unix / network
- Something to do with cloud
- Grew up with Doom, cyberpunk, and the promise of virtual reality

Disclaimers

- Gibson is not supported or endorsed by CMU
- This is an early beta(?)
- I am not an expert at either 3D or Python
- Not all features I'll demonstrate are fully functional

Gibson?

So, what is Gibson?

- A way to model security events in 3D
- Creates a target map (not a network map)
- Highly customizable
- Shows any security alerts as objects that interact with the targets
- Various views allow macro or micro examination
- Pop Up information windows provide specifics
- Visual cues reflect a wide variety of information

Why modeling is useful

- We're visual creatures
- Different people process / learn differently
- Decision makers don't like log files
- An enterprise network has a **lot** of security events.
- Watching people type at the command line is boring

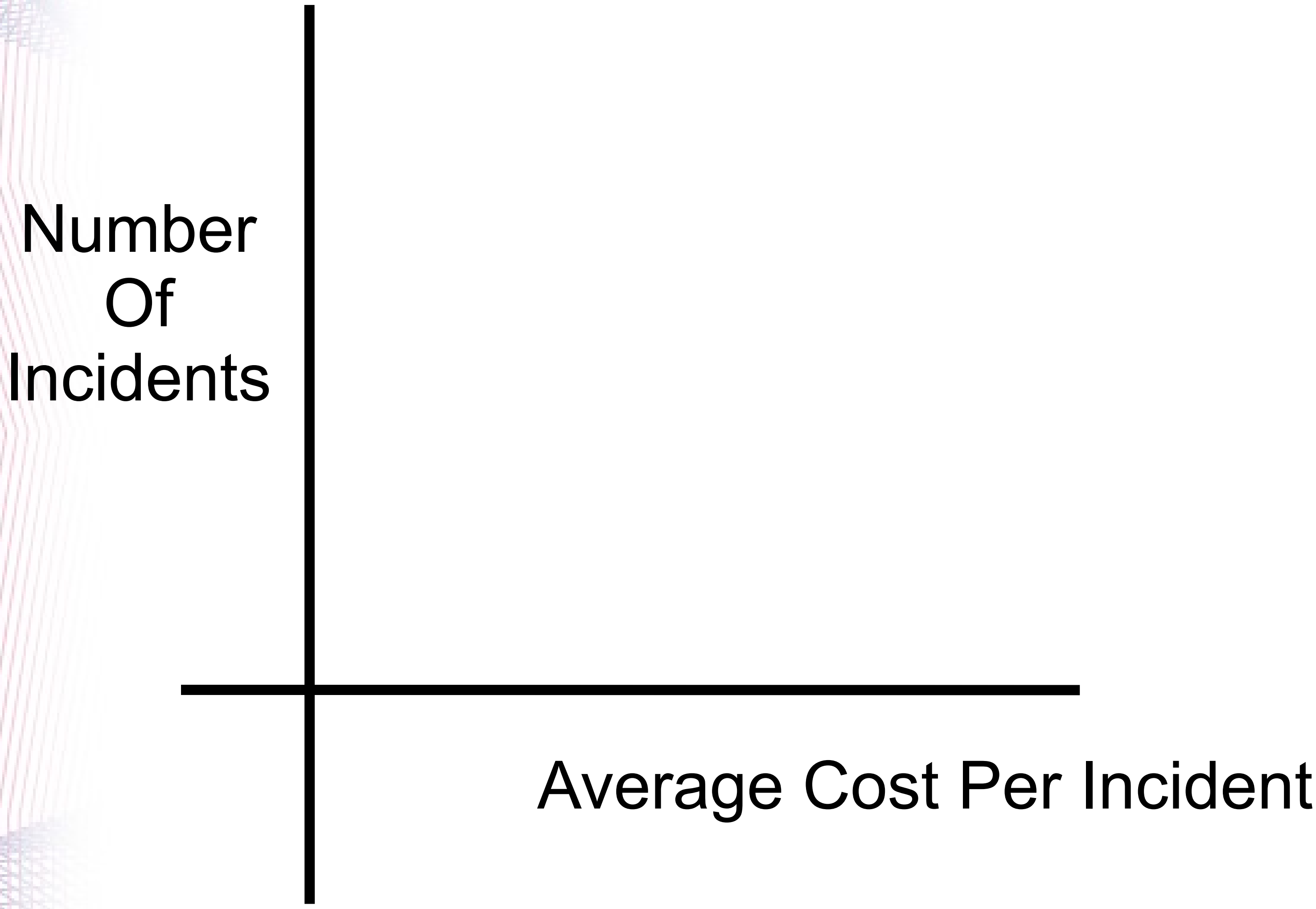
Why real time is useful

- Lots of researchers do studies of archived data
- That's a great way to learn about yesterday's security problems
- Why are exploits called “zero day”?

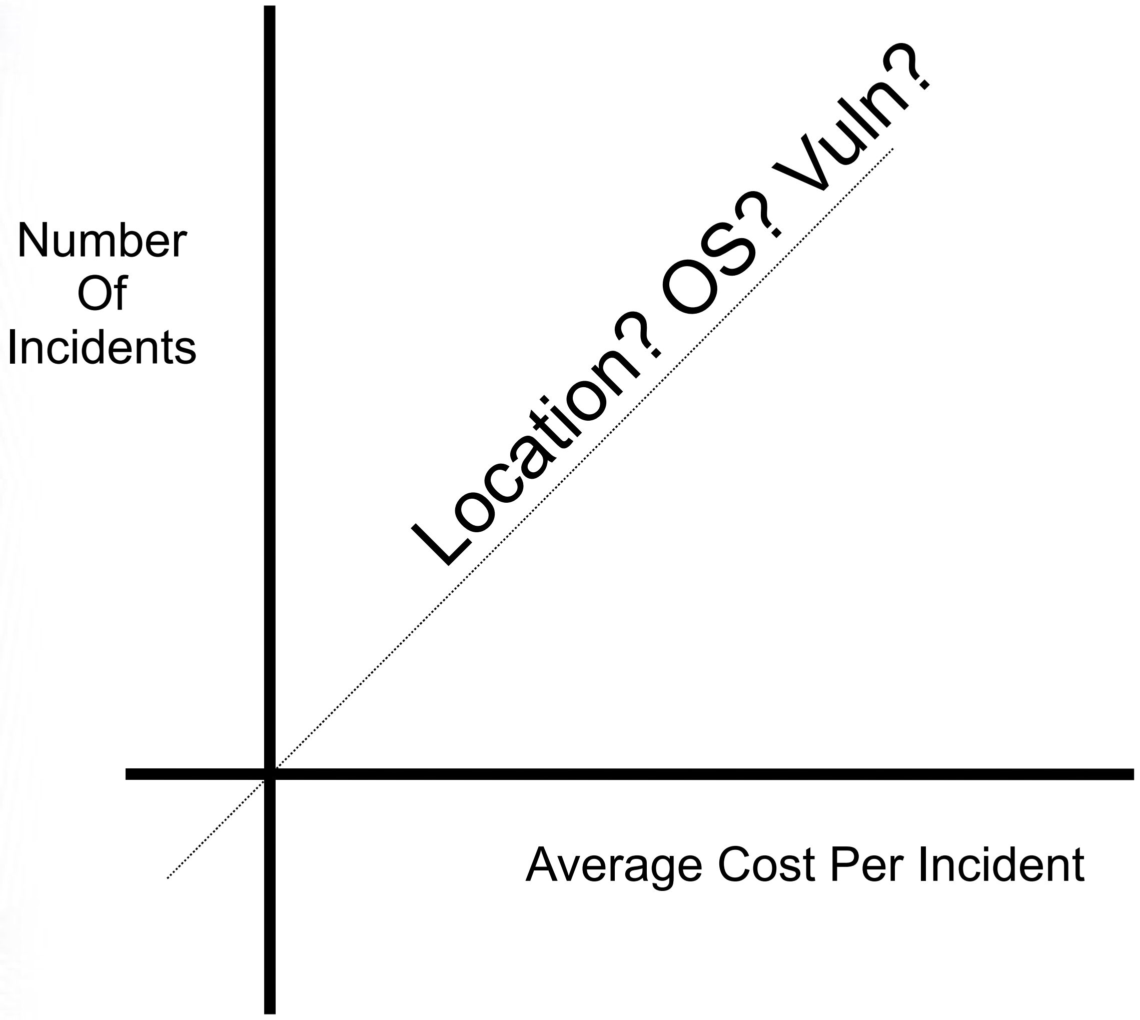
Why historical data is useful

- Research
- Forensics
- Explaining budget request to your boss

Why 3 Dimensions?



Why 3 Dimensions?



The Tech

- Panda3D
  - Created by Disney, now owned / maintained by Carnegie Mellon
  - Better able to control programmatically than others
  - (Relatively) Easy to learn
- Python
- Blender
- Bro (heuristic based IDS)
- Nmap / XML

Let's see it!

We'll start small.

The Basics

- Objects
  - Currently supports nmap XML output as input
  - The main sorting is by IP
    - An IP address seems the best way to identify a target
  - E.g.: Physical or virtual servers, network equip, VPN, other devices
  - Could also be applications, databases, access controls, etc.
  - Could be sorted by OS, function, any custom XML tag
  - You can also build custom models with Python

The Basics

X Axis (Left → Right) = Individual IP Addresses

Z Axis (Up → Down) = Different Subnets

Y Axis (In → Out) = Security Zones

The Basics

- Events
  - Input from monitoring / alarm system
  - Currently: Bro, Snort (fast alerts), and syslog

Bro

```
t=1281415533.93 no=ICMPAsymPayload
na=NOTICE_ALARM_ALWAYS sa=131.243.164.9
sp=60127/tcp da=79.120.86.20 dp=12444/tcp msg=We\ have\
a\ problem tag=@5f-723-2e3d
```

The Basics

Snort

```
11/06/04-01:32:05.706661 {ICMP}
192.168.1.14:3456 - 192.168.100.5:80 TCP [**]
[1:469:3] Bad HTTP [**] [Classification: Attempted
Information Leak] [Priority: 2] [Xref =
http://www.hackers.r.us]
```

Syslog

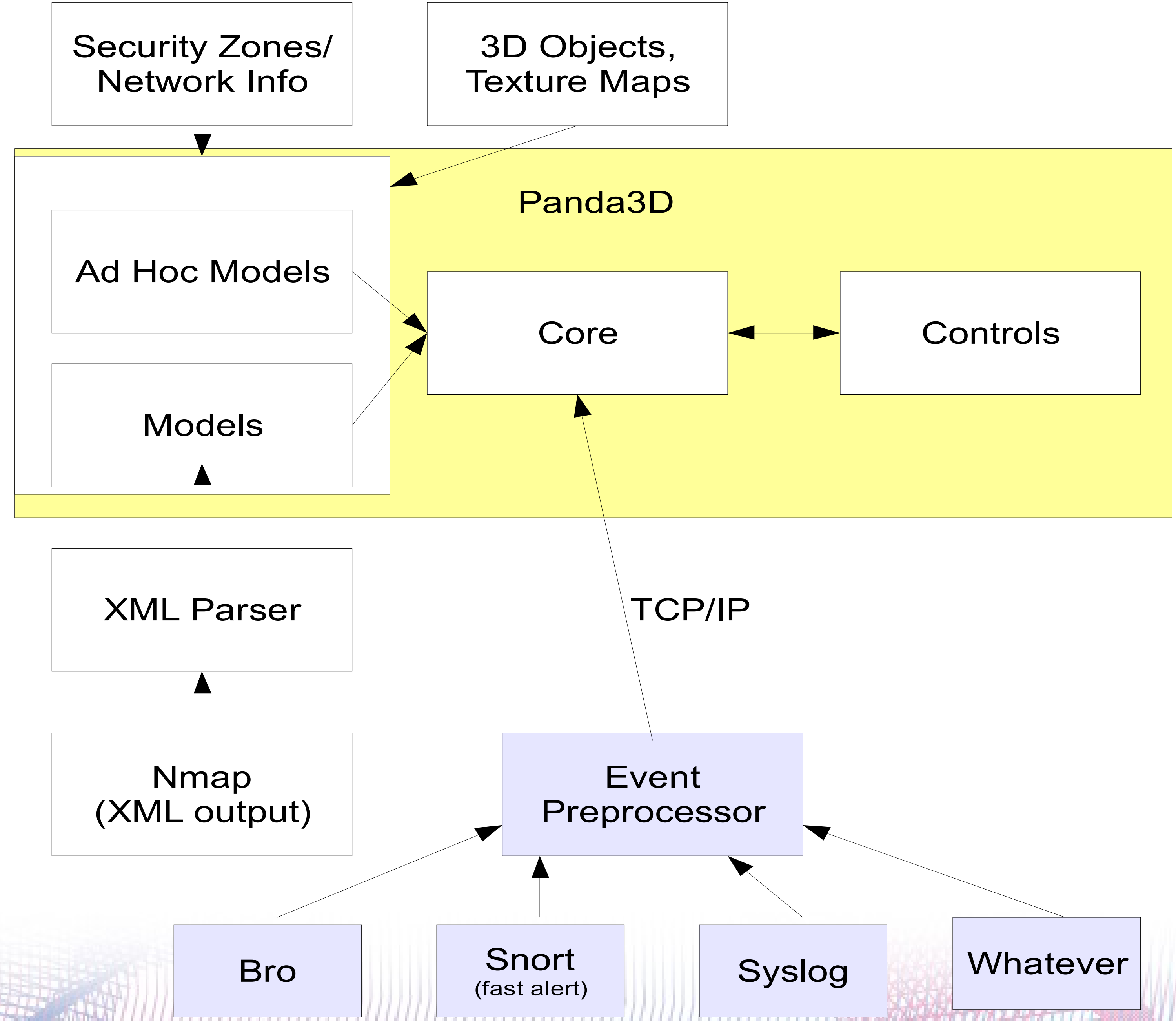
```
Jan  6 13:26:27 132.216.164.24 http-alt[24295]: [ID
800047 local1.error] MyApp: My cool application
done been hacked!
```

The Basics

- Events
  - The Event Processor applies filter you specify
  - Formats for Gibson and Panda
  - Sends to TCP socket in GUI
    - 1283426891.82|BackdoorFound|121.56.72.33|60127/tcp|79.120.86.20|12444/tcp|We have a problem|@5f-723-2e3d

The Basics

- Events
  - Gibson places event in scene depending on various fields
  - Can handle 1,000+ events in the scene; they timeout after user-defined time
  - To replay, just reprocess original logs / events


Security Zones/
Network Info
3D Objects,
Texture Maps
Panda3D
Ad Hoc Models
Core
Controls
Models
XML Parser
TCP/IP
Nmap
(XML output)
Event
Preprocessor
Bro
Snort
(fast alert)
Syslog
Whatever

Panda basics

- Tasks
  - Run every frame, check for mouse clicks, key presses, collisions, etc
- Events
  - Clicks, keys, or user defined
- Animation
  - Intervals – change attributes over time
  - Sequences and Parallels
- Most components are python classes
  - You can inherit from them and extend them

A realistic network, with almost real events

Very small chunk of an enterprise network, with random addresses

Customizing look and feel

- Simple config file

```
[Display]
Skybox = nebula.jpg
```

- It's easy to substitute your own models
- Any large wallpaper will work on the default skybox
- Small textures will work on the skysphere
- Most things inherit from a base color that you can change
- User contributed themes welcome!

The guts

- Very modular
  - I created several different, but fully functional models quickly
- Three base views – Subnet, Single Node, Hybrid
- Each has its own node in Panda, under the root
  - This allows switching and scene-wide change
- The event receiver creates “slugs” for both views
- It's easy to define new slug / tunnel behavior

- Example: Network Clusters with Routing / Proxy

More examples

- These are in various stages of actual functionality

1) Single transaction mode
    1) You could isolate elements that have some shared parameter
2) Whole Subnet (/24) Mode
    1) 256 IP Addresses is too many for one long line
3) Scientific cluster
    1) Very few controllers are gateways to all nodes in the cluster
    2) Nodes are all but identical, no criteria to sort on

What It's Not

- An Intrusion Detection System
- An Event Correlation System
- A Decision Tree
- A Network Map (Yet!)
- A Control Panel (Cannot take action) (Yet!)
- A S(I)(E)M
- Vulnerability Assessment Tool
- A replacement for skilled analysts and auditors

Use Case 1

- Network / Systems Monitoring
- How many NOC Operators still watch logs with tail -f?
- Ability to take in a large amount of aggregated data in a glance
- Ability to explain what's going on without numerous drill downs, reports and graphs

Uh Oh!

Use Case 2

- Vulnerability Assessments / Penetration Tests
- Especially useful if the attackers will allow you to instrument their machines
- See what they're trying, and how you're responding, in real time

Where'd they Get that Exploit?!

Use Case 3

- Simulations / Training Exercises / CTF contests / Test Runs
- Much easier to explain to CEOs, Generals, Politicians & Lawyers

(It doesn't
Do this yet.)

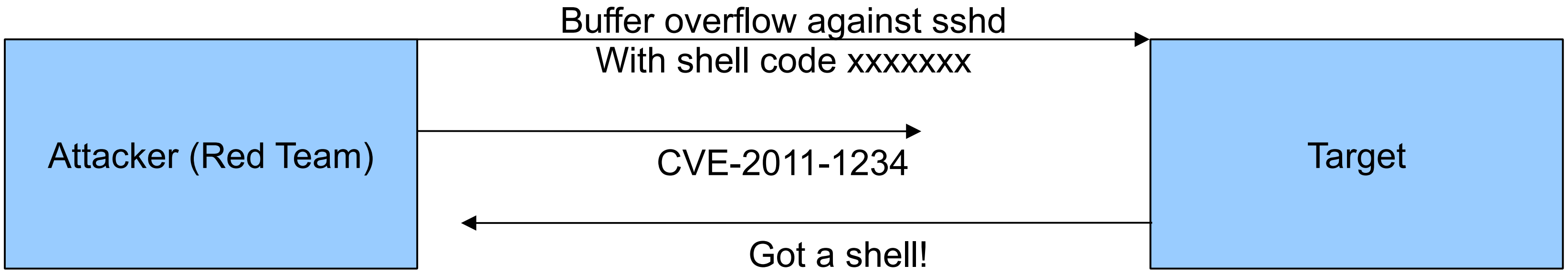
One more thing

Modeling Twitter Relationships

To do list

Host Agents

- An event collector specific to Gibson
- Targets: gather / correlate information
- Attackers: Instrument in simulations



Attacker (Red Team)

Buffer overflow against sshd
With shell code xxxxxxx

CVE-2011-1234

Got a shell!

Target

To do list

Misc. Clean Up

- Better graphic design!
- Code cleanup, error handling, etc.
- Documentation :-(
- GIS / geographic maps
- Model results of vulnerability scans and automated pen test tools

Other uses people have suggested:

1)Model real time processing of AI or expert systems
2)Banking transactions / fraud indicators
3)Represent cyberspace in movies
4)Training classes in networking / security / computers
5)Mapping logical arguments in philosophy(!)

Questions?
Thanks!

Dan Klinedinst
dan@bizling.com